EXPERTENSYSTEME UND IHRE ANWENDUNG AUS BETRIEBSWIRTSCHAFTLICHER SICHT

INAUGURAL - DISSERTATION

zur Erlangung des akademischen Grades
eines Doktors der Wirtschaftswissenschaft
durch die Wirtschaftswissenschaftliche Fakultät
der Westfälischen - Wilhelms - Universität zu Münster

Vorgelegt von

Karl-Ludwig Frhr. von Wendt
aus Bielefeld

1988

UNIV.
LEIDEN
BIBL.

Digitized by the Internet Archive
in 2026

https://archive.org/details/IA41554807_0010

Expertensysteme und ihre Anwendung aus betriebswirtschaftlicher Sicht

Inaugural-Dissertation
zur Erlangung des akademischen Grades
eines Doktors der Wirtschaftswissenschaft
durch die Wirtschaftswissenschaftliche Fakultät
der Westfälischen-Wilhelms-Universität zu Münster

Vorgelegt von
Karl-Ludwig Frhr. von Wendt
aus Bielefeld

1988

D6

1. Gutachter: Prof. Dr. D. Adam

2. Gutachter: Prof. Dr. W. von Zwehl

Dekan: Prof. Dr. H. Bonus

Tag der mündlichen Prüfung: 3. Februar 1988

Inhalt

Seite

Verzeichnis der Abbildungen, Tabellen und Schemata IV
Abkürzungsverzeichnis VI

1 Problemstellung 1
1.1 Zielsetzung der Arbeit 1
1.2 Aufbau der Arbeit 2

2 Darstellung der Expertensystem-Technologie 5
2.1 Begriffliche Grundlagen 5
2.1.1 Begriff und Entwicklung der "Künstlichen Intelligenz" 5
2.1.2 Definition des Begriffs "Expertensystem" 8
2.1.3 Abgrenzung zu konventionellen Programmen 10
2.2 Problemunabhängige Bestandteile eines Expertensystems 14
2.2.1 Das Konzept der Shell 14
2.2.2 Die Inferenzkomponente 16
2.2.3 Die Erklärungskomponente 21
2.2.4 Die Dialogkomponente 23
2.2.5 Die Wissenserwerbs-Komponente 24
2.3 Die Repräsentation von Wissen 26
2.3.1 Produktionssysteme 26
2.3.2 Semantische Netze 28
2.3.3 Frames 29
2.3.4 Der objektorientierte Ansatz 32
2.3.5 Unsicheres Wissen 33
2.4 Hilfsmittel zur Konstruktion von Expertensystemen 41
2.4.1 Vorgefertigte Shells 41
2.4.2 Knowledge-Engineering-Sprachen 44
2.4.3 Allgemeine Programmiersprachen 45

Seite

2.5 Die Expertensystem-Shell MOSES 50
2.5.1 Zielsetzung und Implementierung 50
2.5.2 Wissensdarstellung 52
2.5.3 Die Inferenzkomponente 56
2.5.4 Die Dialogkomponente 60
2.5.5 Die Erklärungskomponente 63
2.5.6 Die Wissensakquisitionskomponente 67
2.5.7 Intern definierte Relationen 68
2.5.8 Kritische Würdigung 70

3 Die Anwendung der Expertensystem-Technologie 72
3.1 Die Erstellung eines Expertensystems 72
3.1.1 Phasen der Erstellung eines Expertensystems 72
3.1.2 Der Prozess der Wissensakquisition 74
3.1.3 Methoden der Wissensakquisition 79
3.1.4 Probleme der Wissensakquisition 84
3.2 Spezialprobleme der Expertensystem-Technologie 89
3.2.1 Das Akzeptanzproblem 89
3.2.2 Das Verantwortlichkeits-Problem 91
3.2.3 Die Bedeutung von Allgemeinwissen für die Problemlösung 93
3.3 Anwendungsbereiche für Expertensysteme 97
3.3.1 Anwendungsvoraussetzungen 97
3.3.2 Problemkategorien 107
3.3.3 Die Vorteilhaftigkeit von Expertensystemen 119
3.4 Anwendungsperspektiven im betriebswirtschaftlichen Bereich 123
3.4.1 Betriebswirtschaftliche Anwendungsmöglichkeiten: Das Beispiel der Digital Equipment Corporation 123
3.4.2 Die Integration in die Anwendungsumgebung 133
3.4.3 Expertensysteme als Beitrag zu einem "Intelligenten Management-Informations-System" 139

Seite

4 FOUNDER: Ein Expertensystem zur Existenzgründungs-Beratung 150
4.1 Existenzgründung als Entscheidungsproblem 150
4.1.1 Problemstellung 150
4.1.2 Der Prozess der Existenzgründung 152
4.1.3 Strategische Entscheidungsbereiche der Existenzgründung 157
4.1.4 Probleme der Existenzgründung 163
4.2 Die Konstruktion eines Expertensystems zur Existenzgründungs-Beratung 168
4.2.1 Aufgabe und Ziel eines Expertensystems zur Existenzgründungs-Beratung 168
4.2.2 Die Struktur der Wissensbasis 171
4.2.3 Die Wissensakquisition 175
4.2.4 Erläuterungen der Regeln 180
4.2.5 Einbindung eines Simulationsmodells zur Liquiditätsprognose 192
4.3 Kritische Würdigung des erstellten Systems 198

5 Ausblick 200

Anhang

A Wissensbasis des Expertensystems FOUNDER 207
B PROLOG-Listing: Die Expertensystem-Shell MOSES 234
C PROLOG-Listing: Simulationsmodell zur Liquiditätsbeurteilung (MOSES-Modul) 249

Literaturverzeichnis 254
Materialverzeichnis 262

Verzeichnis der Abbildungen, Tabellen und Schemata

Abbildungen

		Seite
2.1	Gegenüberstellung von konventionellem Programm und Expertensystem	11
2.2	Aufbau eines Expertensystems	14
2.3	Semantisches Netz	28
2.4	Beispiele für Frames	30
2.5	Beispiele für eine diskrete und eine stetige Zugehörigkeitsfunktion	37
2.6	MOSES-Regel in externer und interner Darstellung	54
2.7	Flußdiagramm der MOSES-Inferenzkomponente	57
2.8	MOSES-Bildschirmaufbau	61
2.9	Warum-Erklärung	63
2.10	Ausschnitt aus einer MOSES-Befragung mit aktiviertem Wie-Trace	66
3.1	Der Prozeß des Knowledge Engineering	73
3.2	Der Prozeß der Wissensakquisition	76
3.3	Problemkategorien	107
3.4	Ausgewählte Expertensysteme bei DEC	127
3.5	Aufbau eines "Intelligenten Management-Informations-Systems"	143
4.1	Der Prozeß der Existenzgründung	153
4.2	Strategische Entscheidungsbereiche der Existenzgründung	158
4.3	Struktur der Wissensbasis des Expertensystems FOUNDER	172
4.4	Programmablaufplan zur Liquiditätssimulation	194

Tabellen

2.1 Beispiel für Zugehörigkeitswerte 39
2.2 Beispiel für die Verknüpfung von Zugehörigkeitswerten 39
3.1 Methoden der Expertenbefragung nach Waterman 83

Schemata

3.1 Vorteilhaftigkeit von Expertensystemen 119
4.1 Berechnung der liquiden Mittel 195

Abkürzungsverzeichnis

Abb.		Abbildung
a.d.	=	außer Dienst
CD	=	Compact Disc
DEC	=	Digital Equipment Corporation
Dir.	=	Direktor
Diss.	=	Dissertation
DM	=	Deutsche Mark
EDV	=	Elektronische Datenverarbeitung
ERP		European Recovery Program
engl.	=	englisch
f.		folgende Seite
ff.	=	fortfolgende Seiten
FGCS	=	Fifth Generation Computer Systems
Gew.	=	Gewerbe
GewO	=	Gewerbeordnung
GmbH	=	Gesellschaft mit beschränkter Haftung
GPS	=	General Problem Solver
GPSS		General Purpose Simulat n System
GuV	=	Gewinn- und Verlustrechnung
H.	=	Heft
Hrsg.	=	Herausgeber
IBM	=	International Business Machines
Jg.	=	Jahrgang
KByte	=	Kilobyte (1024 Bytes)
K.E.	=	Knowledge Engineer
Ltd.	=	Leitender
MOSES	=	Modulare Shell für Expertensysteme
No.	=	Numero
OHG	=	Offene Handelsgesellschaft
o.J.	=	ohne Jahresangabe
o.O.	=	ohne Ortsangabe
PC	=	Personal Computer
ROM	=	Read Only Memory
S.		Seite

Sp. = Spalte
SPSS = Statistical Package for the Social Sciences
s.u. = siehe unten
Tab. = Tabelle
u.a. = und andere
USA = United States of America
USP = Unique Selling Proposition
vgl. = vergleiche
Vol. = Volume

1 Problemstellung

1.1 Zielsetzung der Arbeit

Expertensysteme erregen als zur Zeit bedeutendste praktische Anwendungsform der Methoden der sog. "Künstlichen Intelligenz" in zunehmendem Maße die Aufmerksamkeit der Wissenschaftler verschiedenster Fachrichtungen. Die Vielfalt der bisherigen Anwendungsbereiche für diese Technologie[1] legt die Frage nahe, ob Expertensysteme auch zur Lösung betriebswirtschaftlicher Probleme nutzbringend eingesetzt werden können.

Ziel der vorliegenden Arbeit ist es, dem Wirtschaftswissenschaftler die notwendige Informationsgrundlage zu liefern, um die Anwendbarkeit der Expertensystem-Technologie in seinem Problembereich beurteilen zu können. Weiterhin soll ein Programmsystem entwickelt werden, mit dessen Hilfe betriebswirtschaftliche Expertensysteme auf einfache Weise erstellt werden können. Damit wird dem Wirtschaftswissenschaftler ein Instrument zur Verfügung gestellt, mit dem er den Nutzen der vorgestellten Technologie in seinem Problembereich praktisch erproben kann.

Im theoretischen Teil der Arbeit sollen nach einer Einführung in Aufbau und Funktionsweise von Expertensystemen die Voraussetzungen für eine erfolgreiche Anwendung im betriebswirtschaftlichen Bereich dargestellt und mögliche Probleme aufgezeigt werden. Anhand eines praktischen Beispiels zur Existenzgründung sollen dann die theoretischen überlegungen veranschaulicht werden.

---

1) Waterman nennt z.B. in (91), S.300 ff. allein 16 verschiedene Anwendungsgebiete, darunter Landwirtschaft, Chemie, Computersysteme, Geologie, Rechtswissenschaft, Produktion, Mathematik, Medizin und Weltraumtechnologie.

Aus diesem Grund werden die Aspekte der praktischen Anwendung im betriebswirtschaftlichen Bereich in den Vordergrund gestellt, während die programmtechnischen und informationstheoretischen Aspekte nur insoweit behandelt werden, als es zum Verständnis der praktischen Probleme notwendig ist.

In dieser Arbeit wird bewußt auf den Versuch verzichtet, einen vollständigen überblick über die aktuellen Forschungsarbeiten auf dem Gebiet betriebswirtschaftlicher Expertensysteme zu geben[1]. Obwohl der wissenschaftliche Wert einer solchen Aufstellung nicht bezweifelt werden kann, erscheint sie doch im Rahmen dieser Arbeit nicht zweckmäßig, da sich das Forschungsfeld noch in der Pionierphase befindet[2]. Dies führt zum einen dazu, daß Informationen über die vielfältigen Entwicklungsprojekte schwer zu bekommen sind und zum anderen eine entsprechende Aufstellung bereits wenige Wochen nach ihrer Erstellung überholt wäre[3].

1.2 Aufbau der Arbeit

Aus dem oben gesetzten Ziel läßt sich die Gliederung der Arbeit unmittelbar ableiten. Zunächst sollen im zweiten Kapitel die theoretischen Grundlagen der Expertensystem-Technologie dargelegt werden. Es wird versucht, die Arbeitsweise von Expertensystemen zu veranschaulichen, ohne auf die softwaretechnischen Details der Realisation näher einzugehen. Anhand des im Rahmen dieser Dissertation entwickelten Programmsystems MOSES[4] wird die Funk-

---

1) Entsprechende Aufstellungen finden sich z.B. bei Buchanan (9); Hewett, Timms, d'Annale (35); Lehmann (43); Mertens, Allgeyer, Däs (53); Smart, Langeland-Knudsen (82); Volk (90); Waterman (91), S.239 ff.
2) Vgl. z.B. Volk (90), S.553.
3) Vgl. Mertens, Allgeyer, Däs (53), S.906.
4) MOSES = Modulare Shell für Expertensysteme; vgl. Abschnitt 2.5.

tion einer sog. Expertensystem-Shell als Hilfsmittel zur Konstruktion von Expertensystemen dargestellt.

Das System MOSES kann auch über diese Dissertation hinaus für nachfolgende wissenschaftliche Arbeiten nutzbringend eingesetzt werden[1].

Im dritten Kapitel werden dann die praktischen Probleme der Anwendung von Expertensystemen erläutert. Dabei soll der Leser in die Lage versetzt werden, anhand der Voraussetzungen und dargestellten Schwierigkeiten den Nutzen und die Problematik einer Anwendung der Expertensystem-Technologie in seinem Entscheidungsbereich abschätzen zu können.

Im vierten Kapitel wird ein Prototyp-Expertensystem zur Beratung bei Existenzgründungen vorgestellt, das der Verfasser im Rahmen dieser Arbeit als Demonstrationssystem entwickelt hat. Ziel dieses Systems war es, die Erstellung eines Expertensystems zu erproben und so einen tieferen Einblick in die praktischen Probleme zu gewinnen. Zudem kann an diesem System die prinzipielle Anwendbarkeit von Expertensystemen für betriebswirtschaftliche Fragestellungen gezeigt werden.

Von besonderer Bedeutung scheint dabei, daß die Wissensbasis auf der Befragung eines erfahrenen Experten und nicht allein auf Literaturstudium beruht. Nur auf diese Weise lassen sich nach Meinung des Verfassers die praktischen Probleme der Erfassung und Aufbereitung von Expertenwissen sinnvoll untersuchen. Das System wurde mit Hilfe der Expertensystem-Shell MOSES erstellt.

---

1) Diese Expertensystem-Shell wird inzwischen auch zur Erstellung einer Diplomarbeit am Institut für Betriebswirtschaftslehre, insbesondere Produktion und Absatz der Johann-Wolfgang-Goethe-Universität Frankfurt a.M. eingesetzt; vgl. auch Abschnitt 2.5.

Im fünften Kapitel schließlich soll kurz auf die Bedeutung der Expertensystem-Technologie für die Weiterentwicklung betriebswirtschaftlicher Methoden, insbesondere im Bereich der Unternehmensforschung eingegangen werden. Weiterhin werden die Auswirkungen zukünftiger technischer Entwicklungen auf die Anwendbarkeit von Expertensystemen skizziert.

Ein wesentlicher Teil der wissenschaftlichen Arbeit zu dieser Dissertation war die Entwicklung des Programmsystems MOSES, mit dessen Hilfe betriebswirtschaftliche Expertensysteme einfach und ohne Programmierkenntnisse erstellt werden können, und seine Anwendung bei der Konstruktion eines Expertensystems zur Existenzgründungs-Beratung. Die Ergebnisse dieser Entwicklungsarbeit sind im Anhang in Form von Programmlistings sowie einer Wissensbasis enthalten.

Da sich die vorliegende Arbeit an Wirtschaftswissenschaftler und nicht an Informatiker wendet, wurden die technischen Details und Probleme der Realisation nicht näher beschrieben. Der Umfang der Ausführungen zu den erstellten Systemen entspricht daher nicht dem tatsächlichen Entwicklungsaufwand.

2 Darstellung der Expertensystem-Technologie

2.1 Begriffliche Grundlagen

2.1.1 Begriff und Entwicklung der "Künstlichen Intelligenz"

Das Forschungsgebiet der "Künstlichen Intelligenz" befaßt sich mit dem Versuch, Computerprogramme mit Fähigkeiten, die bei einem Menschen als "intelligent" bezeichnet werden, zu entwickeln[1]. Typische Beispiele für solche Fähigkeiten sind[2]:

- das Verstehen "natürlicher schriftlicher" oder gesprochener Sprache;
- das Erkennen, Verarbeiten und Interpretieren von Bildern;
- das Anwenden von Wissen über den Problembereich (z.B. über typische Problemstellungen und dafür geeignete Lösungsmethoden bzw. Heuristiken) zur Problemlösung;
- das Erkennen der eigenen Unfähigkeit, ein gegebenes Problem zu lösen;
- das Lernen aus Erfahrungen.

Entscheidend ist nicht, daß ein Programm diese Aufgaben auf die gleiche Art und Weise löst wie ein Mensch. Bereits die Tatsache, daß es derartige Probleme lösen kann, rechtfertigt die Bezeichnung "intelligent" im hier gemeinten Sinn.

---

1) Vgl. Waterman (91), S.3. Mit der Definitionsproblematik des Begriffs "Künstliche Intelligenz" beschäftigt sich ausführlich von Zelewski in (99), S.88 ff.
2) Vgl. Savory (74), S.7.

Die obige Interpretation des Begriffs "Künstliche Intelligenz" entspricht dem sog. "Performanzmodus"[1]. Sie hat den Vorteil, daß sie eine Beurteilung der "Intelligenz" eines Computerprogramms anhand objektiver Kriterien ermöglicht. Ihr Nachteil ist, daß sie die Bezeichnung "intelligent" nicht scharf definiert. Dies ist bisher jedoch auch in Bezug auf menschliche Intelligenz noch nicht allgemein anerkannt gelungen[2].

Mit Forschungen zur künstlichen Intelligenz wurde bereits in den fünfziger Jahren begonnen[3]. Anfang der sechziger Jahre wurde der Versuch unternommen, ein "Allgemeines Problemlösungsprogramm", den "General Problem Solver" (GPS), zu entwickeln[4]. Dieses Programm sollte jedes denkbare Problem durch Anwenden grundlegender Problemlösungsmethoden lösen können. Der Versuch mußte jedoch bald als gescheitert angesehen werden. Der überhöhte Anspruch des GPS sowie die Fehlinterpretationen der ersten Versuche zum Verständnis natürlicher Sprache, die durch Weizenbaums Programm "Eliza"[5]

---

1) Im "Performanzmodus" wird die "Intelligenz" eines Computerprogramms ausschließlich anhand seiner Leistungen beurteilt. Es ist dabei unerheblich, auf welche Weise diese erbracht werden. Vgl. von Zelewski (99), S.127 ff.
2) Vgl. z.B. Meili (50), Sp.997 ff.
3) Als "Geburtsstunde der künstlichen Intelligenz" wird die sog. Dartmouth Conference 1956 bezeichnet, auf der die Grenzen des neuen Forschungsgebietes abgesteckt wurden. Vgl. von Zelewski (99), S.31.
4) Vgl. Rich (71), S.9 ff.; Yuhan (97).
5) ELIZA simuliert einen Psychiater, indem es Schlüsselwörter aus den Eingaben des Benutzers herausgreift und darauf aufbauend Sätze bildet. Es verfügt über keinerlei semantisches Verständnis des eingegebenen Textes, sondern erzeugt lediglich die Illusion von Intelligenz - dies allerdings so überzeugend, daß sich zahlreiche Benutzer in den sechziger Jahren lieber mit ELIZA als mit einem menschlichen Psychiater über ihre Probleme unterhalten wollten. Vgl. Weizenbaum (92), S.36 ff.

ausgelöst wurden, brachten der künstlichen Intelligenz den Ruf der Unseriosität, von dem sie sich erst Mitte der siebziger Jahre befreien konnte.

Bereits gegen Ende der sechziger Jahre erkannte man, daß Probleme nicht unabhängig von den speziellen Eigenschaften des Problembereichs gelöst werden können[1]. Man konzentrierte sich nun auf eng begrenzte Problemfelder und versuchte, die dort auftretenden Probleme mit Wissen über das jeweilige Fachgebiet zu lösen. Eines der ersten Ergebnisse dieses Umdenkens und gleichzeitig eines der ersten Expertensysteme war das System DENDRAL zur Interpretation von Daten eines Massenspektrometers[2].

Mitte der siebziger Jahre folgten weitere Expertensysteme wie MYCIN[3] zur Diagnose von Infektionskrankheiten und PROSPECTOR[4] zur Aufspürung von Bodenschätzen anhand von geologischen Daten, die spektakuläre Erfolge für sich verbuchen konnten. In einem Test, in dem die Leistungen von MYCIN denen von Fachärzten gegenübergestellt wurden, erreichte das Expertensystem die mit Abstand geringste Fehlerquote[5]. PROSPECTOR entdeckte bei einem praktischen Test ein Molybdän-Lager von beachtlichem Wert[6].

Die beschriebenen Erfolge scheinen in heutiger Zeit zu einer gewissen Expertensytem-Euphorie geführt zu haben, wie aus den zahlreichen unkritischen Artikeln in Fach-

---

1) Vgl. Waterman (91), S.4.
2) Vgl. Buchanan, Feigenbaum (10), S.5 ff.
3) Vgl. z.B. Shortliffe (79).
4) Vgl. Gaschnig (26).
5) Vgl. Gaschnig u.a. (27), S.264 f.
6) Vgl. Campbell u.a. (13), S.927 ff.

und Publikumszeitschriften zu entnehmen ist[1]. Dies ist bedenklich, da übersteigerte Erwartungen wie bereits in den sechziger Jahren enttäuscht werden und die künstliche Intelligenz erneut in Mißkredit bringen könnten.

2.1.2 Definition des Begriffs "Expertensystem"

In der Literatur findet sich keine einheitliche Definition des Begriffs "Expertensystem"[2]. Stattdessen findet man vielfach Beschreibungen der typischen Eigenschaften eines Expertensystems, die je nach Betrachtungsstandpunkt unterschiedlich ausfallen[3].

Der Begriff "Expertensystem" entspringt dem Wunsch, die Fähigkeiten eines menschlichen Experten auf einem Computer nachzuvollziehen[4]. Damit würde die Definitions-Problematik auf den Begriff des "Experten" verschoben. Man ist sich in der Literatur jedoch weitgehend einig, daß die in der Praxis einsetzbaren Expertensysteme menschliche Experten nicht voll ersetzen können und sollen[5]. Es ist daher für diese Arbeit zweckmäßig, die Definition des Begriffs "Expertensystem" nicht an einem bestimmten Leistungsanspruch, sondern an der Funktionsweise dieser Systeme zu orientieren. Aus diesem Grund soll hier die folgende, eher technische Definition verwendet werden:

---

1) Als krasses Beispiel mag die folgende Schlagzeile einer Meldung in der Tageszeitung "Westfälische Nachrichten", Münster, vom 9.3.1987 dienen: "Der Computer macht es möglich: Patient stellt seine Diagnose selbst". Vgl. auch die ausführliche Darstellung dieser Problematik bei von Zelewski (99), S.22 ff.
2) Vgl. Buchanan (9), S.32.
3) Vgl. z.B. Waterman (91), S.XVII, Jackson (39), S.1, Harmon, King (33), S.3, Savory (74), S.23.
4) Vgl. Puppe (68), S.1.
5) Vgl. z.B. Luconi, Malone, Morton (45), S.6, Sieben, Böing, Hafner (80), S.535, Waterman (91), S.13 f.

"Expertensysteme sind Computerprogramme, die einen problemunabhängigen Schlußfolgerungsmechanismus auf eine problemspezifische Wissensbasis anwenden, um die für die Lösung eines Problems notwendigen Informationen zu extrahieren."

Diese Definition deckt sich im wesentlichen mit dem in der Literatur verwendeten Begriff "Wissensbasiertes System" bzw. "Knowledge-Based System"[1]. Sie ist weiter gefaßt als viele in der Literatur zu findende Definitionen[2], da sie nicht den Anspruch einer bestimmten Leistungsfähigkeit umfaßt.

Der Begriff "Expertensystem" nach obiger Definition umfaßt damit nicht nur Systeme, die Laien das Wissen von Experten zur Verfügung stellen sollen, sondern auch solche, die den Experten selbst bei der Arbeit unterstützen und entlasten. Diese Systeme werden in der Literatur teilweise als "Experten-Unterstützungs-Systeme" (expert support systems) bezeichnet[3].

Expertensysteme in diesem Sinne können neben den Methoden der künstlichen Intelligenz auch konventionelle, numerisch orientierte EDV-Verfahren wie etwa Simulation oder statistische Analysen zur Lösungsfindung einsetzen. Gerade diese Eigenschaft dürfte bei Anwendungen im betriebswirtschaftlichen Bereich eine besondere Rolle spielen.

---

1) Vgl. z.B. Waterman (91), S.18, Savory (74), S.24
2) Vgl. z.B. Waterman (91), S.7 und S.11.
3) Vgl. Luconi, Malone, Morton (45), S.6, ähnlich: Ernst, Ojha (20), S.173.

Der Grund für die weite Fassung des Begriffs liegt darin, daß Expertensysteme in der Praxis nur sehr selten die Ansprüche erfüllen können, die an einen menschlichen Experten gestellt werden[1]. Die Entwicklung geht daher von der klassischen Vorstellung eines Systems, das den menschlichen Experten ersetzen soll, hin zu dem Ziel, Systeme zu schaffen, die Experten bei ihrer Arbeit unterstützen und sie von lästigen Routineaufgaben entlasten sollen[2].

2.1.3 Abgrenzung gegenüber konventionellen Programmen

Aus der angeführten Definition ergibt sich die strikte Trennung zwischen wenigstens einer problemunabhängigen Komponente, nämlich dem Schlußfolgerungsmechanismus, und der problemspezifischen Wissensbasis. Diese Trennung ist das wichtigste Kennzeichen wissensbasierter Systeme und gleichzeitig ein geeigneter Ansatz zur Abgrenzung gegenüber herkömmlichen Programmen[3]. Abbildung 2.1[4] soll diesen Unterschied verdeutlichen.

---

1) Vgl. z.B. Bell (3), S.618.
2) Vgl. Luconi, Malone, Morton (45), Ernst, Ojha (20).
3) Vgl. Schmitz, Lenz (77), S.505.
4) Vgl. Puppe (68), S.1.

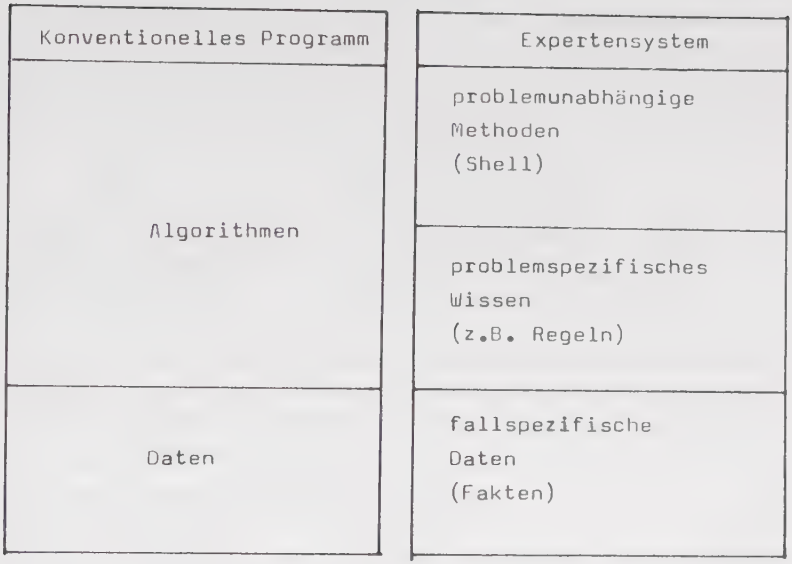


Abb. 2.1: Gegenüberstellung von konventionellem Programm und Expertensystem

Jedes in irgendeiner Form nützliche Computerprogramm enthält implizit Wissen über das Problem, das es lösen soll, und die Methoden zu seiner Lösung[1]. Ein Programm zur Berechnung des Kapitalwertes einer Investition z.B. enthält Wissen darüber, wie die verschiedenen Input Größen zueinander in Beziehung zu setzen sind, damit als Resultierende der Kapitalwert berechnet werden kann.

1) Vgl. Schmitz, Lenz (77), S.505.

Dieses Wissen kann die Form von Fakten, d.h. Wissen über reale Gegebenheiten, oder Prozeduren, d.h. Wissen über die anzuwendenden Methoden, haben. Entsprechend dem Konzept über den Aufbau eines Computers nach von Neumann[1] wird ersteres als Daten, letzteres als Algorithmen bezeichnet. Im obigen Beispiel wären Daten z.B. das Anfangskapital, die Dauer der Investition oder der zugrundegelegte Kalkulationszinsfuß. Der Algorithmus wäre z.B. die klassische Kapitalwert-Formel.

Bei einem konventionellen EDV-Programm ist selbst in einem Programm-Listing nicht ohne weiteres zu erkennen, welche Programmteile problemspezifische und welche problemunabhängige Algorithmen (wie etwa die Fähigkeit zur Errechnung eines Mittelwertes) enthalten.

Im Unterschied zum klassischen von-Neumann-Konzept lassen sich Expertensysteme in drei unabhängige Teile einteilen. Den Algorithmen in konventionellen Programmen stehen zum einen problemunabhängige Lösungsmethoden, zum anderen problemspezifisches Expertenwissen gegenüber. Das Expertenwissen umfaßt dabei sowohl Fakten als auch prozedurales Wissen. Den Daten in konventionellen Programmen kann man schließlich die Fakten über ein konkretes Problem gegenüberstellen, die in der Regel während der Konsultation eines Expertensystems vom Benutzer angegeben oder von externen Datenbeständen eingelesen werden.

---

1) Vgl. z.B. von Zelewski (99), S.314.

Die explizite Darstellung des bereichsspezifischen Wissens in Expertensystemen hat gravierende Auswirkungen auf die Flexibilität und Leistungsfähigkeit dieser Systeme und ist ein entscheidender Grund für den Erfolg dieses Konzeptes. Sie ermöglicht es,

Erklärungskomponenten[1] zu konstruieren, die in der Lage sind, die einzelnen Schritte einer Schlußfolgerung offen darzulegen, und die somit die Fehlersuche in der Wissensbasis sehr erleichtern und die Glaubwürdigkeit der Systeme wesentlich erhöhen;

- die Wissensbasis sehr leicht veränderten Problembedingungen anzupassen bzw. das gespeicherte Wissen schrittweise zu erweitern und zu korrigieren[2];

- problemunabhängige Rahmenprogramme, sogenannte Shells[3] (engl: Schale, Hülle) zu konstruieren, die die Komponenten zur Darstellung und Verarbeitung des Wissens enthalten und die Konstruktion von Expertensystemen in den verschiedensten Anwendungsbereichen sehr vereinfachen.

---

1) Vgl. Abschnitt 2.2.3.
2) Vgl. Abschnitt 3.1.1.
3) Vgl. den folgenden Abschnitt.

2.2 Problemunabhängige Bestandteile eines Expertensystems

2.2.1 Das Konzept der Shell

Abbildung 2.2 zeigt die einzelnen Bestandteile eines typischen Expertensystems im Überblick[1]. Wie bereits erläutert enthält die "Shell" die problemunabhängigen Komponenten, die in den folgenden Abschnitten näher beschrieben werden, während in der Wissensbasis das problemspezifische Wissen enthalten ist.

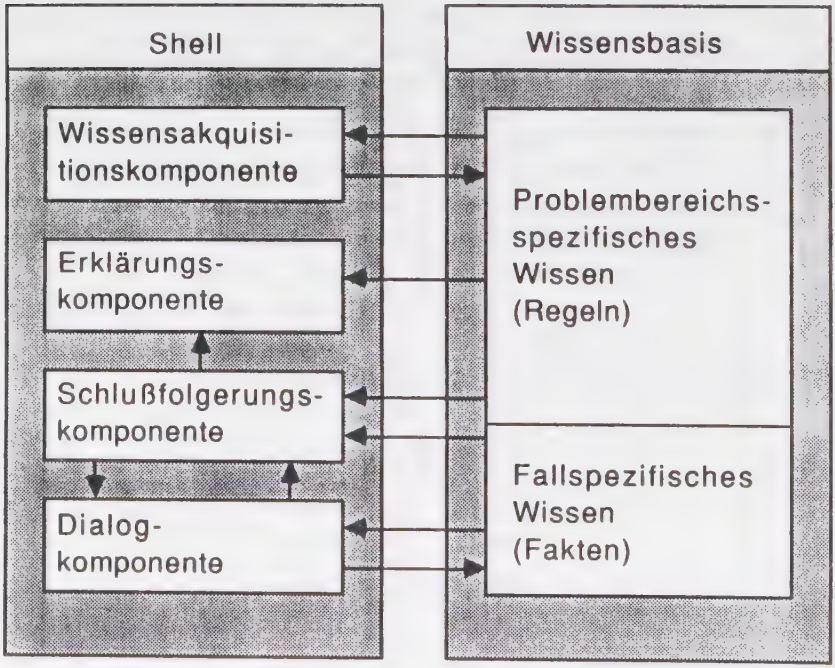


Abb. 2.2: Aufbau eines Expertensystems

1) Vgl. z.B. Puppe (68), S.2, Waterman (91), S.19.

Dieses umfaßt sowohl allgemeines Wissen über den Problembereich, das z.B. in Form von Regeln gespeichert werden kann, als auch Wissen über den konkreten Anwendungsfall. Letzteres wird während einer Konsultation vom Benutzer eingegeben oder automatisch, z.B. von Meßgeräten oder Datenbanken, erfaßt. Beide Komponenten sind logisch und programmtechnisch klar voneinander getrennt.

Wie weit diese Trennung in der Praxis geht, zeigt das Beispiel des Expertensystems MYCIN[1]. Dieses System wurde konstruiert, um infektiöse Krankheiten zu diagnostizieren. Später wurde die Wissensbasis entfernt und man erhielt die Shell EMYCIN[2], was für "Empty MYCIN" steht. Mit Hilfe von EMYCIN wurden dann u.a. die Expertensysteme PUFF zur Diagnose von Lungenkrankheiten und LITHO zur Analyse von Gesteinsformationen konstruiert[3].

Der wesentliche Vorteil des Konzeptes der Shell besteht darin, daß die einmal konstruierten problemunabhängigen Bestandteile ohne Änderungen zur Erstellung von Expertensystemen in den verschiedensten Anwendungsbereichen benutzt werden können. Dies bedeutet eine erhebliche Arbeitsersparnis, und es ermöglicht die Erstellung der Wissensbasis auch ohne ein Beherrschen der Programmiertechniken der künstlichen Intelligenz. Allerdings gibt es verschiedene Techniken der Repräsentation und Verarbeitung von Wissen, so daß eine Vielzahl unterschiedlicher Shells existiert, die sich für verschiedene Problemstellungen unterschiedlich gut eignen[4].

---

1) Vgl. z.B. Jackson (39), S.93 ff.
2) Vgl. ebenda, S.107 ff.
3) Vgl. Waterman (91), S.114.
4) Vgl. Abschnitt 2.4.1.

Bei den in der Praxis erstellten, als "Expertensysteme" bezeichneten Programmen ist die Trennung zwischen Shell und Wissensbasis allerdings nicht immer eindeutig. So ist es im Extremfall auch möglich, ein System, das sich wie ein Expertensystem verhält, in einer "konventionellen" Programmiersprache wie etwa BASIC zu schreiben[1] und so die Abgrenzung zwischen den problemunabhängigen Komponenten und der Wissensbasis zu verwischen. Die Frage, ob ein derartiges System noch als Expertensystem zu bezeichnen ist, ist oft schwierig zu beantworten[2]. Nach der in Abschnitt 2.1.2 gegebenen Definition muß zumindest eine logische Trennung beider Komponenten gegeben sein, die den schrittweisen Ausbau der Wissensbasis[3] und die Erklärung der Schlußfolgerungen erlaubt.

2.2.2 Die Inferenzkomponente

Kern der Shell ist die Inferenzkomponente, die auch als Schlußfolgerungskomponente oder Inferenzmaschine ("Inference Engine") bezeichnet wird[4]. Diese Komponente hat die Aufgabe, aus den in der Wissensbasis vorhandenen Fakten, Regeln und Beziehungen sowie den vom Benutzer eingegebenen Daten neue Fakten zu erschließen.

So kann z.B. aus den Fakten

"Ein Auto hat vier Räder" und
"Ein VW-Käfer ist ein Auto" die Aussage
"Ein VW-Käfer hat vier Räder" geschlossen werden.

Diese Schlußfolgerung mag trivial erscheinen, aber bei einer Größenordnung von einigen hundert bis einigen

1) Vgl. Abschnitt 2.4.3.
2) Vgl. Mertens, Allgeyer, Däs (53), S.905.
3) Vgl. Abschnitt 3.1.1.
4) Vgl. z.B. Waterman (91), S.289.

tausend Regeln, wie sie bei Expertensystemen in der Praxis vorkommen[1], sind die resultierenden Schlußfolgerungen nicht immer so offensichtlich.

Die meisten Inferenzkomponenten sind in der Lage, unsicheres bzw. unscharfes Wissen zu verarbeiten. So ist z.B. der Satz "Ein Auto hat meistens vier Räder" eine unscharfe Aussage, da der Begriff "meistens" nicht mehr mit den Begriffen "wahr" oder "falsch" ausgedrückt werden kann. Wenn die Aussage "Ein VW-Käfer ist ein Auto" wahr ist, so ist die Schlußfolgerung "Ein VW-Käfer hat vier Räder" nun nicht mehr mit Sicherheit wahr, sondern mit einem Sicherheits- oder Wahrscheinlichkeitswert behaftet. Auf die Möglichkeiten zur Verarbeitung unscharfen Wissens wird in Abschnitt 2.3.5 näher eingegangen.

Bei der Schlußfolgerung auf der Basis des vorhandenen Wissens kann zwischen zwei Strategien unterschieden werden, die oft auch miteinander kombiniert werden. Um diese zu erläutern soll der Begriff der "Produktionsregel"[2] als eine Möglichkeit der Darstellung von Wissen eingeführt werden. Auf diese und andere Methoden wird im Abschnitt 2.3 detailliert eingegangen.

Unter einer Produktionsregel versteht man eine Aussage der Form "WENN Bedingung_1 und Bedingung_2 und... Bedingung_n DANN Folgerung"[3]. Eine Produktionsregel (im folgenden vereinfacht Regel genannt) ist dann erfüllt, d.h. die Aussage im Folgerungs-Teil ist dann wahr, wenn alle Bedingungen 1 bis n erfüllt sind. Haben mehrere Regeln denselben Folgerungsteil, so genügt zur Bestätigung der Folgerung die Erfüllung des Bedingungsteils nur einer Regel.

1) Vgl. z.B. Schieferle (75), S.56.
2) Vgl. z.B. Coombs (16), S.29, Harmon, King (33), S.30, Jackson (39), S.31 ff.
3) Produktionsregeln können auch umgekehrt die Form "DANN Folgerung WENN Bedingungen" haben.

Die gebräuchlichste Schlußfolgerungsstrategie ist die sog. Rückwärtsverkettung ("Backward Chaining")[1]. Dabei wird, ausgehend von einem gegebenen Ziel wie etwa der Beantwortung einer Benutzer-Frage, zunächst eine Regel ausgewählt, die in ihrem Folgerungsteil das angestrebte Ziel enthält. Dann wird versucht, alle im Bedingungsteil dieser Regel enthaltenen Aussagen zu beweisen, indem diese nun ihrerseits zu Zielen erhoben und Regeln gesucht werden, die diese Aussagen bestätigen.

Bei der vorwärtsgerichteten Schlußfolgerung bzw. Vorwärtsverkettung ("Forward Chaining")[2] wird eine Regel immer dann angewendet, wenn ihr Bedingungsteil erfüllt ist, unabhängig davon, ob die resultierende Folgerung in Beziehung zu der eigentlichen Fragestellung steht oder nicht. Diese Strategie führt oft zu ineffizienten bzw. irrelevanten Schlußfolgerungen, weshalb sie als alleinige Strategie selten verwandt wird[3].

Die Arbeitsweise einer "regelbasierten Inferenzkomponente" soll an dem folgenden Beispiel verdeutlicht werden, wobei die Großbuchstaben für beliebige Aussagen stehen. Gegeben sei folgender Satz von Produktionsregeln:

1) Wenn A und B dann C
2) Wenn C und E dann F
3) Wenn E und G dann H
4) Wenn A, F und D dann G
5) Wenn A und G dann I

1) Vgl. Waterman (91), S.68 f.
2) Vgl. ebenda, S.66 f.
3) Vgl. ebenda, S.67.

In der Datenbasis mögen die Fakten A, B, C, D und E bereits enthalten sein (z.B. weil der Benutzer diese zuvor angegeben hat), d.h. es steht fest, daß diese Aussagen wahr sind. Das Ziel sei es, die Aussage H zu bestätigen oder zu verwerfen.

Eine rückwärtsverkettete Inferenzmaschine würde dazu folgendermaßen vorgehen: da H noch nicht in der Datenbasis vorhanden ist, würde zunächst eine Regel gesucht, die in ihrem Folgerungsteil die gesuchte Aussage enthält. Im Beispiel ist dies Regel Nr.3. Es werden nacheinander die Bedingungen dieser Regel zu Zielen erhoben und zu bestätigen versucht, d.h. das System sucht zunächst in der Datenbasis, ob E bereits vorhanden ist. Da dies der Fall ist, wird G zum Ziel erhoben. Diese Aussage ist noch nicht in der Datenbasis enthalten, aber Regel Nr.4 enthält G im Folgerungsteil. Also werden nacheinander A, F und D zu Zielaussagen usw. Das System kann schließlich die Aussage H durch Anwenden der Regeln 3, 4 und 2 bestätigen.

Eine vorwärtsverkettete Inferenzmaschine geht in der umgekehrten Reihenfolge vor. Zunächst wird versucht, Regel 1 anzuwenden. Dies gelingt, da A und B in der Datenbasis enthalten sind, führt jedoch zu keiner zusätzlichen Information, da auch die Folgerung C bereits bekannt ist. Danach wird Regel 2 angewandt und F in die Datenbasis geschrieben. Regel 3 kann nicht angewandt werden, da G noch nicht bestätigt ist. Regel 4 und 5 führen dann zur Aufnahme von G und I in die Datenbasis, da alle ihre Voraussetzungen erfüllt sind. Schließlich wird die Regelbasis ein zweites Mal durchlaufen, wobei solche Regeln, die bereits erfolgreich angewandt wurden, nicht mehr berücksichtigt werden. Dies führt zur Anwendung von Regel 3 und damit der Bestätigung von H. Die vorwärtsverkettete Inferenzkomponente wendet dazu also die Regeln 1, 2, 4, 5 und 3 an.

Vergleicht man die beiden Vorgehensweisen, so erkennt man, daß die rückwärtsverkettete Methode im Beispiel effizienter ist. Die Regeln 1 und 5, die keinen unmittelbar nutzbringenden Informationszuwachs bewirken, werden im Gegensatz zur vorwärtsverketteten Methode nicht angewandt. Für die rückwärtsverkette Methode ist es jedoch notwendig, daß man die genaue Fragestellung zu Beginn kennt, d.h. man muß im Beispiel vorher wissen, daß H die gesuchte Aussage ist. Ist dies nicht der Fall, müssen alle denkbaren Lösungen als Zielaussagen nacheinander geprüft werden; die rückwärtsverkettete Methode ist dann nicht mehr notwendig die effizientere Vorgehensweise.

In vielen Fällen ist es sinnvoll, beide Vorgehensweisen miteinander zu kombinieren. Dies kann z.B. geschehen, indem ein Satz vorwärtsverketteter Regeln auf eine Datenbasis angewandt wird, um Hypothesen über mögliche Lösungen aufzustellen. Jede Hypothese wird dann mit rückwärtsverketteten Regeln getestet, bis eine der Hypothesen bestätigt ist. Diese Vorgehensweise wird als "Hypothesize-and-Test"-Strategie[1] bezeichnet. Sie ist vor allem sinnvoll, wenn eine große Zahl möglicher Lösungen existiert, es aber anhand von Indizien in der Datenbasis möglich ist, diese auf wenige "aussichtsreiche" Lösungskandidaten zu reduzieren.

---

1) Vgl. Puppe (68), S.5.

2.2.3 Die Erklärungskomponente

Ein wesentliches Merkmal von Expertensystemen ist die Fähigkeit, ihre Schlußfolgerungen erklären bzw. begründen zu können[1]. Die Erklärungskomponente muß zwei wichtige Aufgaben erfüllen:

Zum einen bildet die Erklärungsfähigkeit die Grundlage für das Vertrauen des Benutzers in das Expertensystem[2]. Sie ermöglicht es, die Schlußfolgerungs-Ketten des Systems nachzuvollziehen, und macht sie somit verifizierbar. Zum anderen ist die Erklärungskomponente ein wichtiges Instrument zur Lokalisierung und Beseitigung von Fehlern in der Wissensbasis. Unzulässige Schlüsse können vom Experten rasch erkannt und die Ursachen erschlossen werden.

Der Nutzen der Erklärungskomponenten beim heutigen Stand der Technik ist jedoch eingeschränkt[3]. Die bisher entwickelten Expertensysteme können in der Regel nur die zur Erreichung einer Schlußfolgerung angewandten Regeln und Fakten aufzählen. Die Schlußfolgerung ist damit nur nachvollziehbar, wenn der Benutzer über genügend Fachwissen verfügt, um die verwendeten Regeln zu verstehen und die Richtigkeit der Schlüsse beurteilen zu können.

Eine Erklärungskomponente der beschriebenen Art erfüllt nicht die Anforderungen, die an menschliche Experten gestellt werden. Diese sollen nämlich den Abstraktionsgrad ihrer Erklärungen dem Wissensstand des Fragenden anpassen[4]. Inwieweit menschliche Experten gewillt oder in der Lage sind, diesen Anforderungen nachzukommen, sei dahingestellt.

---

1) Vgl. Waterman (91), S.28 f.
2) Vgl. Swartout (86), S.298.
3) Vgl. Puppe (68), S.8.
4) Vgl. ebenda.

Die Erklärungskomponenten einzelner Expertensysteme bzw. Shells variieren sowohl hinsichtlich ihrer Methodik als auch ihrer Leistungsfähigkeit stark. Einfachere Systeme können lediglich die Regeln aufzeigen, die sie zur Erlangung einer Schlußfolgerung angewandt haben[1], während komplexere Systeme in natürlichsprachlichem Dialog mit dem Benutzer die Hintergründe einer Frage oder Schlußfolgerung erläutern[2].

Man kann zwei grundlegende Typen von Erklärungen unterscheiden[3]. Zum einen gibt es die sog. "Warum"-Erklärung. Wenn der Benutzer eine Frage des Systems beantworten soll, so kann er durch Eingabe des Wortes "Warum" oder eines ähnlichen Schlüsselwortes eine Erklärung dafür verlangen, warum gerade diese Frage zu diesem Zeitpunkt gestellt wird. Das System gibt dann meist die Regel an, deren Anwendung die Frage ausgelöst hat. Oft besteht auch die Möglichkeit, den Grund für die Anwendung eben dieser Regel weiter zurückzuverfolgen.

Die "Warum"-Erklärung eröffnet dem Benutzer, sofern er über entsprechendes Fachwissen verfügt, einen Einblick in den Kontext der Frage, so daß er sie unter Umständen präziser und sachgerechter beantworten kann. Außerdem zeigt sie, an welcher Stelle des Schlußfolgerungsprozesses sich das System zur Zeit befindet.

---

1) So z.B. die in Abschnitt 2.5 beschriebene, vom Verfasser erstellte Expertensystem-Shell MOSES.
2) Vgl. z.B. das Beispiel PROSPECTOR in Waterman (91), S.50 ff.
3) Vgl. Hammond (32), S.306 ff.

Die zweite Form ist die sog. "Wie"-Erklärung. Sie kann die Frage beantworten, wie ein bestimmtes Ergebnis erreicht werden konnte, bzw. unter Umständen auch, warum es nicht erreicht wurde. Dazu wird eine Regel agegeben, deren Folgerungsteil die gestellte Frage beinhaltet, und es wird angegeben, ob die einzelnen Bedingungen für die Regel erfüllt wurden. Bei Bedarf kann dann weiterverfolgt werden, wie einzelne Bedingungen bestätigt wurden, indem auch dafür die angewandten Regeln gezeigt werden.

Die "Wie"-Erklärung dient primär dem Ziel der Fehlersuche bei der Erstellung der Wissensbasis. Aus den oben erwähnten Gründen wird sie dem Anspruch, die Schlußfolgerungen für den Nicht-Experten verständlich begründen zu können, kaum gerecht. Dennoch haben Expertensysteme selbst mit derart einfachen Erklärungskomponenten gegenüber konventionellen Programmen einen großen Vorsprung in Bezug auf Benutzerfreundlichkeit und Nachvollziehbarkeit[1].

2.2.4 Die Dialogkomponente

Die Dialogkomponente bildet die Schnittstelle zwischen Benutzer und Expertensystem. Durch sie kann der Benutzer den Ablauf der Konsultation steuern, Fragen beantworten oder die Erklärungskomponente aktivieren. Eine komfortable und flexible, d.h. in Bezug auf Eingabefehler oder problemspezifische Besonderheiten (z.B. Abkürzungen) tolerante Dialogkomponente ist eine wichtige Voraussetzung für die Akzeptanz eines Expertensystems[2].

---

1) Vgl. Puppe (68), S.8.
2) Vgl. ebenda, S.7.

Im einzelnen muß die Dialogkomponente den Zugang zu den Funktionen des Expertensystems ermöglichen, Fragen des Systems an den Benutzer oder Hinweis- bzw. Erklärungstexte ausgeben und Antworten des Benutzers einlesen. Dabei müssen die Eingaben des Benutzers auf Richtigkeit (z.B. Eingabe innerhalb des zulässigen Wertebereichs) geprüft werden; Abkürzungen oder Umschreibungen (z.B. Zahlworte oder unscharfe Begriffe wie "vielleicht") müssen in ein vom Expertensystem zu verarbeitendes Format umgewandelt werden. Einige Expertensysteme verfügen auch über Möglichkeiten der graphischen Darstellung, z.B. in Form von Anzeigeinstrumenten[1].

Ein ideales Expertensystem wäre in der Lage, mit dem Benutzer in natürlicher Sprache zu kommunizieren. Die Verwendung von Fachausdrücken und die Ausführlichkeit von Fragen wären dem Wissensstand des Benutzers angepaßt, so daß der Fachmann nicht durch zu langwierige Erklärungen in seinem Gedankenfluß gehemmt, der Laie wiederum nicht durch unverständliche Fragen frustriert würde. Leider sind jedoch die heutigen Expertensysteme von derartigen Fähigkeiten noch weit entfernt[2].

2.2.5 Die Wissenserwerbs-Komponente

Die Wissenserwerbs- oder Wissensakquisitions-Komponente hat die Aufgabe, den Ersteller der Wissensbasis, sei er ein Fachgebiets-Experte oder ein sog. Knowledge Engineer[3], bei der Aufbereitung, Eingabe und Verbesserung des Wissens zu unterstützen.

---

1) Vgl. Waterman (91), S.101.
2) Vgl. Puppe (68), S.7.
3) Vgl. Abschnitt 3.1.3.

Meist wird das Wissen in einem Expertensystem intern in einem relativ unverständlichen oder unübersichtlichen Format gespeichert[1]. Es ist daher hilfreich, wenn die Wissenserwerbs-Komponente über die "Fähigkeit" verfügt, Regeln, Fakten oder Relationen in einer der natürlichen Sprache ähnlichen Form zu erfassen und sie dann in das interne Format zu übersetzen.

Weiterhin gehört es zu den Aufgaben der Wissensakquisitions-Komponente, die Struktur der Wissensbasis zu veranschaulichen und so Änderungen und Fehlersuche zu erleichtern. Allerdings verfügen nicht alle Expertensysteme bzw. Shells über derartige Möglichkeiten. Eine leistungsfähige Wissenserwerbskomponente kann die Entwicklung eines Expertensystems wesentlich beschleunigen.

Eine für Wissenserwerbs-Komponenten oft geforderte Eigenschaft ist die Fähigkeit zur Konsistenzprüfung. Dabei soll das Expertensystem in der Lage sein, widersprüchliches Wissen in der Wissensbasis zu erkennen und somit die Fehlersuche zu erleichtern. Die Forschung auf diesem Gebiet hat jedoch bisher noch keine in der Praxis anwendbaren Ergebnisse gebracht[2].

---

1) Vgl. Abb. 2.6 in Abschnitt 2.5.2.
2) Vgl. Waterman (91), S.92.

2.3 Die Repräsentation von Wissen

2.3.1 Produktionssysteme

Unter einem Produktionssystem versteht man eine Menge von Produktionsregeln der im Abschnitt 2.2.2 beschriebenen Form, ergänzt um einen Regel-Interpretierer und einen Arbeitsspeicher, in dem Zwischenergebnisse abgelegt werden[1]. Dabei können einige der Regeln Bedingungen enthalten, die dem Folgerungsteil anderer Regeln entsprechen. Auf diese Weise lassen sich Schlußfolgerungsketten ("Inference Chains")[2] herstellen, mit denen aus einer Menge von gegebenen Fakten neue Fakten "produziert", d.h. gefolgert werden können. Das Produktionssystem als Methode der Repräsentation von Wissen hat gegenüber den anderen Methoden einige wesentliche Vorteile:

- Produktionsregeln sind sehr einfach strukturiert und damit z.B. auch von Experten, die mit den Methoden der Künstlichen Intelligenz nicht vertraut sind, unmittelbar zu verstehen. Oft ist die Produktionsregel die "natürliche" Form, um Expertenwissen, das in Form von Heuristiken bzw. "Faustregeln" vorliegt, zu formulieren[3].

- Produktionssysteme sind programmtechnisch relativ leicht zu realisieren und bieten sich damit vor allem für Experimente und Prototypen, die der Erforschung der Anwendbarkeit von Expertensystemen dienen, an. Mit Hilfe der Programmiersprache PROLOG, deren Programme prinzipiell als Produktionssysteme angesehen werden können[4], lassen sich nach kurzer Einarbeitungszeit kleine Wissensbasen erstellen.

---

1) Vgl. z.B. Harmon, King (33), S.30, Jackson (39), S.31.
2) Vgl. Waterman (91), S.65 f.
3) Vgl. ebenda, S.63.
4) Vgl. von Zelewski (99), S.210.

- Produktionsregeln haben einen selbst-erklärenden Charakter, d.h. für eine einfache Erklärungskomponente genügt es oft, wenn die jeweils verwendete Regel ohne weiteren Begleittext dargestellt wird.

- Produktionssysteme sind die am weitesten verbreitete Methode der Wissensdarstellung in Expertensystemen[1]. Aus diesem Grunde gibt es für diesen Ansatz die meisten Hilfsmittel[2] und Erfahrungsberichte.

Der wesentliche Nachteil der Verwendung von Produktionssystemen liegt in der resultierenden unübersichtlichen und schlecht strukturierten Wissensbasis. Dies läßt sich mit der Unübersichtlichkeit eines Programms in einer schwach strukturierten Programmiersprache wie etwa BASIC im Gegensatz zu einem Programm in der wohl strukturierten Sprache Pascal vergleichen. In einer großen Wissensbasis läßt sich praktisch nicht mehr überblicken, welche Konsequenzen das Entfernen oder Ändern einer Regel für das Verhalten des Expertensystems haben wird.

Ein weiterer Nachteil des Produktionssystem-Ansatzes liegt in seiner Ineffizienz. Da dieser Ansatz keine Möglichkeit bietet, Elemente der Wissensbasis zu Gruppen oder Klassen zusammenzufassen, die sich durch gleichartige Merkmale auszeichnen, ist die Darstellung von hierarchisch strukturiertem Wissen, wie es etwa in wissenschaftlichen Taxonomien vorkommt, vergleichsweise umständlich[3].

---

1) Vgl. Waterman (91), S.63.
2) Vgl. die Aufstellungen in Waterman (91), S.105 und 107 f.
3) vor allem im Vergleich zur Methode der sog. Frames, die in Abschnitt 2.3.3 dargestellt wird.

2.3.2 Semantische Netze

Eine weitere Form der Repräsentation von Wissen sind die sog. Semantischen Netze[1]. Darunter versteht man eine Menge von sinntragenden Symbolen bzw. Begriffen, die durch Beziehungen miteinander verbunden sind. Abb. 2.3 veranschaulicht ein einfaches Semantisches Netz.

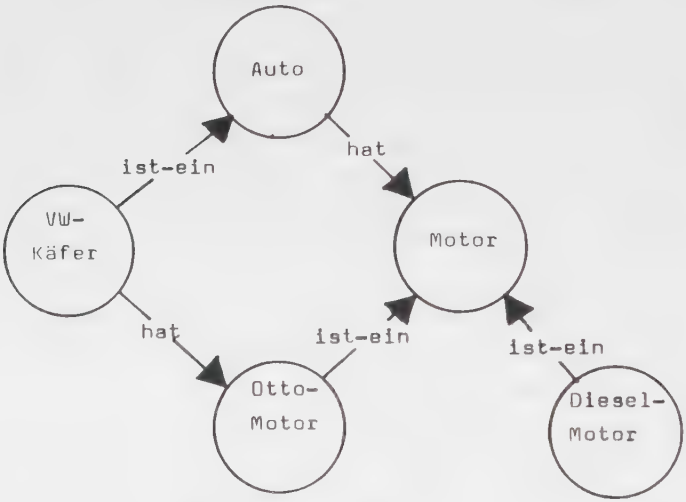


Abb. 2.3: Semantisches Netz

Im Rahmen der Expertensystem-Technologie sind die Beziehungen "ist-ein" und "hat" von besonderer Bedeutung. Mit ihrer Hilfe lassen sich hierarchisch strukturierte Wissensbasen aufbauen[2]. Aus den Relationen

"VW Käfer ist ein - Auto" und

"Auto - hat - Motor"

läßt sich, ähnlich wie im regelbasierten System, folgern, daß ein VW-Käfer einen Motor hat.

---

1) Vgl. Sowa (83), S.1011 ff.
2) Vgl. ebenda, S.1016 f.

Semantische Netze dienen dazu, Begriffshierarchien zu formulieren. Ihre besondere Stärke liegt in der Möglichkeit der graphischen Darstellung des Beziehungsgeflechts in einer Wissensbasis, die allerdings in der Praxis aufgrund der Menge des darzustellenden Wissens schnell an ihre Grenzen stößt. Semantische Netze haben vor allem im Bereich des Verstehens natürlicher Sprache große Bedeutung erlangt. Als Methode zur Repräsentation von Wissen innerhalb eines Expertensystems sind sie vor allem in solchen Problembereichen geeignet, in denen es um die Identifikation von Objekten oder Sachverhalten geht[1].

2.3.3 Frames

Der Ansatz der sog. Frames (engl.: Rahmen) als Methode der Repräsentation von Wissen erlangt zunehmende Bedeutung in der Konstruktion von Expertensystemen. Unter einem Frame versteht man eine Datenstruktur, die einen bestimmten Begriff, eine Situation oder ein Objekt bzw. eine Kategorie von Objekten beschreibt. Jeder Frame enthält dazu verschiedene sog. Slots (engl.: Schlitz, hier sinngemäß etwa: Schubfach), in denen Informationen über den beschriebenen Begriff gespeichert sind[2]. Abb. 2.4 zeigt ein Beispiel für Frames[3].

Ein wesentliches Merkmal der Wissensrepräsentation durch Frames ist das Konzept der Vererbung[4]. Es lassen sich zu jedem Frame Unterkategorien bilden, deren Elemente als Spezifikationen des allgemeineren Frames angesehen werden können. Alle Elemente der Unterkategorie

---

1) Zu den verschiedenen Problemkategorien vgl. Abschnitt 3.3.1.
2) Vgl. z.B. Maida (46), S.302 ff.
3) Vgl. Koch, Heuer (41), S.7 f. Das Beispiel ist hier vereinfacht wiedergegeben.
4) Vgl. Koch, Heuer (41), S.9.

"erben" sämtliche Informationen aller ihnen direkt hierarchisch übergeordneten Frames. Dabei können allerdings Merkmale, die von dem übergeordneten Frame als "Vorgabewerte" ("default values") ererbt worden sind, durch speziellere Ausprägungen überschrieben werden[1].

```
KLASSE Wertpapier
SLOTS
  Risiko (niedrig/mittel/hoch)
  Land (Symbol)
  börsengängig (ja/nein)


UNTERKLASSE Kursneutrales_Anlagepapier
IST_EIN Wertpapier MIT
  Risiko = niedrig
  börsengängig = nein
SLOTS
  Rendite (Real)
  Nennwert (Integer)
  Rückzahlungstermin (Datum)
  Laufzeit (Integer)


UNTERKLASSE Bundesschatzbrief
IST_EIN Wertpapier MIT
  Land = Bundesrepublik Deutschland
IST_EIN Kursneutrales_Anlagepapier
SLOTS
  Typ (A/B)
  Nominalzinsen (Liste von Realzahlen)
  Verkauf_ab (Datum)
  Kenn_Nr (Integer)

INSTANZ BSB-A-1985/11
IST_EIN Kursneutrales_Anlagepapier MIT
  Nennwert = 100 DM
  Laufzeit = 6 Jahre
  Rendite = 5.84 %
  Rückzahlungstermin = 1.9.1991
IST_EIN Bundesschatzbrief MIT
  Typ = A
  Kenn_Nr = 110747
  Verkauf_ab = 26.8.1985
  Nominalzinsen = (4%, 5%, 6%, 6.25%, 6.5%, 8%)
```

Abb. 2.4: Beispiele für Frames

1) Vgl. Koch, Heuer (41), S.9.

Hat z.B. der Frame "Wertpapier" einen Slot "börsengängig", in dem per Vorgabewert die Ausprägung "ja" enthalten ist, so erben alle Unterkategorien diese Information, es sei denn, sie wird explizit anders angegeben. Der Frame "Bundesschatzbrief" könnte z.B. die Information "ist ein Wertpapier mit börsengängig = nein" enthalten. Der Bundesschatzbrief hätte damit alle Eigenschaften des Frame "Wertpapier" mit den entsprechenden Vorgabewerten übernommen. Lediglich die Börsengängigkeit wäre für das spezielle Modell neu definiert worden.

Slots können neben den entsprechenden Informationen auch prozedurales Wissen darüber enthalten, was zu tun ist, wenn die Information des betreffenden Slots benötigt wird, aber nicht vorhanden ist. In diesem Fall könnte z.B. eine Frage an den Benutzer formuliert oder das Ergebnis durch Berechnung aus anderen Slots gewonnen werden. Man spricht von sog. "Demons" (engl.: Dämonen), die "gerufen" werden, um die entsprechende Information zu beschaffen[1]. Diese Prozeduren können z.B. die Form von Produktionssystemen haben.

Die besondere Stärke des Konzepts der Frames ist die Möglichkeit, Objekte zu Klassen bzw. Gruppen zusammenzufassen, die gleichartige Eigenschaften aufweisen. Da ein Objekt alle Eigenschaften jeder Klasse, der es angehört, erbt, brauchen diese Eigenschaften nur einmal nämlich in der entsprechenden Klasse - definiert zu werden. Auf diese Weise kann hierarchisch strukturiertes Wissen sehr effizient formuliert werden. Der Nachteil dieser Methode liegt darin, daß sie programmtechnisch relativ aufwendig zu realisieren ist.

---

1) Vgl. Maida (46), S.307.

2.3.4 Der objektorientierte Ansatz

Ein neuerer Ansatz der Repräsentation von Wissen ist der sog. "objektorientierte Ansatz"[1]. Dabei handelt es sich allerdings eher um ein generelles Prinzip der Programmierung als nur um eine Methode zur Darstellung von Wissen. Der objektorientierte Ansatz kann als eine Verallgemeinerung des Konzepts der Frames angesehen werden[2]. Wie Frames haben auch Objekte bestimmte Eigenschaften und gehören übergeordneten Kategorien als spezielle Ausprägungen an bzw. haben wiederum Unterkategorien von Objekten als Spezialisierungen.

Anders als bei den Frames gibt es jedoch keinen übergeordneten Steuerungsmechanismus, der die Auswertung des Wissens kontrolliert. Stattdessen können Objekte als unabhängige Einheiten angesehen werden, die über ihnen zugeordnete Eigenschaften, Informationen und Prozeduren verfügen. Benötigt ein Objekt eine Information von einem anderen Objekt, so "schickt" es eine entsprechende Anforderung an dieses Objekt, das wiederum dem anfordernden Objekt das Ergebnis mitteilt.

Es würde den Rahmen dieser Darstellung sprengen, das Konzept der objektorientierten Programmierung ausführlich und verständlich darzustellen. Man könnte das Prinzip dieser Methode modellhaft mit einer Konferenz vergleichen, auf der sich verschiedene Spezialisten miteinander unterhalten und versuchen, die Informationen der anderen jeweils durch ihr eigenes Fachwissen anzureichern, um so schließlich ein gemeinsames Problem zu lösen. Tatsächlich erinnert der Name der wichtigsten objektorientierten Programmiersprache, SMALLTALK[3] (engl.: Konversation, Plauderei), an dieses Bild.

---

1) Vgl. Liebermann (44).
2) Vgl. Maida (46), S.309.
3) Vgl. Geller (28); Liebermann (44), S.455.

Dem objektorientierten Ansatz kommt eine ähnliche Bedeutung zu wie den oben beschriebenen Frames. Man könnte ihn als eine grundlegende Technik beschreiben, um das Konzept der Frames bereits auf der Ebene der Programmierung zu verwirklichen. Entsprechend hat er die selben Vor- und Nachteile als Instrument zur Repräsentation von Wissen wie die Frames. Hinzu kommt, daß der Ansatz relativ jung ist und entsprechend noch keine weitreichenden Erfahrungen bei der Konstruktion von Expertensystemen vorliegen. Als leistungsfähiges, flexibles Konzept der Programmierung könnte der Ansatz jedoch in der Zukunft große Bedeutung im Bereich der Künstlichen Intelligenz und der Expertensysteme erlangen.

2.3.5 Unsicheres Wissen

Ein wesentliches Problem nahezu aller Methoden zur Entscheidungsfindung ist die Tatsache, daß die entscheidungsrelevanten Informationen in der Praxis meist stochastische Größen sind[1]. Dies kann sich auf Daten oder Problemzusammenhänge beziehen. So ist es denkbar, daß von einer Inputgröße für ein mathematisches Modell nur der Mittelwert und näherungsweise eine Verteilungsform geschätzt werden können, oder daß Zusammenhänge zwischen einzelnen Größen des Modells nicht genau bekannt sind. Entsprechend sind auch die Ergebnisse derartiger Modelle stochastische Größen; ihre Interpretation ist teilweise problematisch.

---

1) Vgl. Adam (1), S.87.

Die Unsicherheit des Wissens in einem Expertensystem kann prinzipiell vier verschiedene Quellen haben[1]. Zum einen kann die Information als solche unsicher sein, wie dies z.B. bei der Wettervorhersage der Fall ist, d.h. der Wahrheitsgehalt einer Information läßt sich nicht objektiv bestimmen. Weiterhin kann die Information unvollständig sein, d.h. der Wahrheitsgehalt ist zwar theoretisch bestimmbar, die dafür benötigten Daten sind jedoch nicht verfügbar. Eine dritte Quelle der Unsicherheit resultiert aus dem Informationsverlust, der durch Datenaggregation entsteht.

Die vierte Ursache für die Unsicherheit von Wissen ist die Unschärfe bzw. fehlende Präzision der menschlichen Sprache. Sie äußert sich in Begriffen wie "groß" oder "deutlich größer als". Sie ist besonders im Bereich der Anwendung von Expertensystemen bedeutungsvoll, weil solche Formulierungen in Heuristiken und Erfahrungsregeln, wie sie Bestandteil des Expertenwissens sind, sehr häufig vorkommen.

Um unsichere bzw. unscharfe Aussagen in einem Modell zu verarbeiten, ist es notwendig, ihnen exakte Aussagen zuzuordnen, die die Art der Unsicherheit möglichst gut beschreiben. Dies geschieht in mathematischen Modellen in der Regel durch Zuordnung von Wahrscheinlichkeiten. So ist die Aussage "Ein Auto hat mit einer Wahrscheinlichkeit von 0,95 vier Räder" der Versuch, die Unschärfe der Aussage "Ein Auto hat meistens vier Räder" in einer exakten, mathematisch verwendbaren Aussage zu formulieren. Wahrscheinlichkeiten verketteter Aussagen können nach den Regeln der Statistik verknüpft werden[2].

---

1) Vgl. Bouissone (8), S.854.
2) Vgl. z.B. Bleymüller, Gülicher, Gehlert (6), S.29 ff.

Für die Verwendung in Expertensystemen erscheint das Rechnen mit Wahrscheinlichkeits-Werten jedoch nur bedingt geeignet. Zum einen ist in der Praxis die Voraussetzung der statistischen Unabhängigkeit[1] der zu untersuchenden Aussagen nicht gewährleistet. Zum anderen ist die Berechnung von bedingten Wahrscheinlichkeiten jedoch in der Praxis oft unmöglich oder zu aufwendig, da die stochastischen Abhängigkeiten von Ereignissen oder Sachverhalten nicht quantifiziert werden können. Abstrahiert man dagegen von den tatsächlichen Abhängigkeiten und unterstellt statistische Unabhängigkeit, so täuscht die Exaktheit des wahrscheinlichkeitstheoretischen Ansatzes über diese heuristische Vorgehensweise hinweg[2].

Aus diesem Grund wird in den meisten Expertensystemen auf die Berechnung von Wahrscheinlichkeiten verzichtet, und es werden stattdessen verschiedene heuristische Ansätze verwendet. Diese Ansätze arbeiten in der Regel mit sog. Wahrheits- oder Glaubwürdigkeitswerten[3], die nicht (wie Wahrscheinlichkeitswerte) den Anspruch eines objektiven Wertes erheben, sondern subjektiv bzw heuristisch bestimmt werden.

An dieser Stelle kann nicht auf alle in Expertensystemen verwendeten Ansätze zur Verarbeitung unscharfen Wissens eingegangen werden. Stattdessen soll ein Konzept vorgestellt werden, das auf der "Theorie der unscharfen Mengen" aufbaut[4]. Dieses Konzept erscheint

1) Vgl. Bleymüller, Gülicher, Gehlert (6), S.34
2) Vgl. Ganoe (25), S.33.
3) Im Original "certainty factors" oder "confidence factors", vgl. z.B. Waterman (91), S.22
4) Im Original "Fuzzy Sets"; vgl. Bellmann, Zadeh (4), Zadeh (100).

für die Zwecke dieser Arbeit besonders geeignet, da die Theorie der unscharfen Mengen bereits im Rahmen der Unternehmensforschung Anwendung gefunden hat[1] und dieser Ansatz die Unschärfe der menschlichen Sprache besonders gut abbilden kann[2].Er wurde auch bei der Konstruktion der Expertensystem-Shell MOSES zugrundegelegt[3].

Während in der Booleschen Algebra nur die Wahrheitswerte "wahr" und "falsch" zugelassen sind, die oft durch die Zahlenwerte 0 und 1 ausgedrückt werden, erlaubt die Theorie der unscharfen Mengen auch alle Ausprägungen zwischen diesen beiden Werten. Ein "VW-Käfer" gehört also nicht mehr <u>entweder</u> in die Menge der "schnellen" Autos <u>oder</u> in die Komplementärmenge der "nicht-schnellen" Autos. Stattdessen erhält er zu beiden Mengen einen Zugehörigkeitswert, der ausdrücken soll, "wie stark" ein Objekt zu einer Menge gehört, d.h. im Beispiel, wie gut die Bezeichnung "schnell" auf das Objekt "VW-Käfer" zutrifft[4].

Zur Veranschaulichung könnte man sich vorstellen, daß der Zugehörigkeitswert angibt, wie wahrscheinlich es ist, daß das Objekt von einer beliebigen Person der jeweiligen Gruppe zugeordnet wird. In der Praxis wird der Zugehörigkeitswert bzw. die Zugehörigkeitsfunktion jedoch meist von einer einzelnen Person subjektiv[5] bestimmt, so daß diese Interpretation lediglich bildhaften Charakter hat.

---

1) Vgl. z.B. Zimmermann (101), S.785 ff.
2) Vgl. Ganoe (25), S.34.
3) Vgl. Abschnitt 2.5.2.
4) Vgl. Negoita (59), S.50 ff.
5) Vgl. ebenda, S.51.

Zugehörigkeitswerte sind grundsätzlich heuristischer Natur. Für jede unscharfe Menge muß eine Zugehörigkeitsfunktion[1] gebildet werden, die es erlaubt, jedem beliebigen Element einen Zugehörigkeitswert zuzuordnen. Abb. 2.5 zeigt zwei Beispiele für derartige Zugehörigkeitsfunktionen. Sie können stetige oder diskrete Form haben. Dabei gibt Z (G) den Zugehörigkeitswert eines Autos zur unscharfen Menge der "schnellen" Autos in Abhängigkeit von seiner Höchstgeschwindigkeit G in km/h an.

$$Z(G) = \begin{cases} 0 & \text{für } G =< 120 \\ 0{,}2 & \text{für } 120 < G =< 140 \\ 0{,}4 & \text{für } 140 < G =< 160 \\ 0{,}6 & \text{für } 160 < G =< 170 \\ 0{,}8 & \text{für } 170 < G < 180 \\ 1 & \text{für } G >= 180 \end{cases}$$

$$Z(G) = \begin{cases} 0 \text{ für } G <= 130 \\ 1 - ((180 - G)/50) \text{ für } 130 < G < 180 \\ 1 \text{ für } G >= 180 \end{cases}$$

Abb. 2.5: Beispiele für eine diskrete und eine stetige Zugehörigkeitsfunktion

Für die Verknüpfung unscharfer Mengen finden üblicherweise die folgenden Regeln Anwendung[2]:

- Der Durchschnitt (Konjunktion) mehrerer unscharfer Mengen wird gebildet, indem jedem Element das Minimum der Zugehörigkeitswerte zugeordnet wird.
- Die Vereinigung (Disjunktion) mehrerer unscharfer Mengen wird gebildet, indem jedem Element das Maximum der Zugehörigkeitswerte zugeordnet wird.

---

1) Vgl. Negoita (59), S.50 f.
2) Vgl. Zimmermann (101), S.787.

Die Komplementärmenge einer unscharfen Menge wird gebildet, indem jedem Element die Differenz zwischen dem maximalen Zugehörigkeitswert (meist 1) und dem Zugehörigkeitswert der Ursprungsmenge zugeordnet wird.

Die genannten drei Regeln bilden nicht die einzige Möglichkeit der Verknüpfung unscharfer Mengen. So wäre etwa auch die multiplikative Verknüpfung im Falle der Konjunktion denkbar[1]. Ob das hier vorgestellte Verknüpfungsprinzip sinnvoll ist, hängt dabei von der Struktur des zu lösenden Problems ab.

In dem Expertensystem FOUNDER, das auf diesem Prinzip aufbaut, wird z.B. die Existenzgründung als nicht erfolgversprechend angesehen, wenn nur eine der zahlreichen Voraussetzungen nicht erfüllt ist, unabhängig davon, wie gut die übrigen Bedingungen erfüllt sind. Die Verknüpfung unscharfer Mengen nach den oben genannten Regeln erscheint damit vor allem für solche Probleme geeignet, bei denen eine "vorsichtige" Beurteilung angebracht ist, wie dies bei der Beurteilung einer Existenzgründung zweifellos der Fall ist.

Nicht geeignet ist diese Methode dagegen, wenn nicht nur der minimale Wahrheitswert, sondern auch die Wahrheitswerte der übrigen Bedingungen zur Beurteilung einer Aussage berücksichtigt werden sollen.

Dies wäre im Beispiel der Existenzgründung dann sinnvoll, wenn man nicht wissen möchte, ob ein vorliegendes Gründungskonzept erfolgversprechend ist, sondern wieweit es von einem Idealkonzept abweicht. Das Ergebnis der Beurteilung würde etwas darüber aussagen, wieviel

---

1) Vgl. Zimmermann (101), S.794 f.

Verbesserungen an dem Konzept notwendig wären. Es wäre in diesem Fall jedoch denkbar, daß ein relativ gut beurteiltes Gründungskonzept in der Praxis an einer einzigen, nicht erfüllten Bedingung scheitert.

Die dargestellten Regeln zur Verknüpfung unscharfer Mengen sollen an einem Beispiel verdeutlicht werden. Gegeben seien folgende Zugehörigkeitswerte verschiedener Auto-Typen zu den Mengen der "schnellen", "geräumigen" und "preiswerten" Autos:

Auto-Typ	Porsche 911	Mercedes 200D	VW-Käfer
"schnell"	1	0,5	0
"geräumig"	0	0,6	0,1
"preiswert"	0	0,4	1

Tab. 2.1: Beispiel für Zugehörigkeitswerte

Aus diesen Angaben lassen sich entsprechend den genannten Regeln folgende verknüpfte Zugehörigkeitswerte ermitteln:

Auto-Typ	Verknüpfungsart	Porsche 911	Mercedes 200D	VW - Käfer
"schnell und preiswert"	Minimum	0	0,4	0
"geräumig oder preiswert"	Maximum	0	0,6	1
"nicht preiswert"	1- Zugehörigkeitswert	1	0,6	0

Tab. 2.2: Beispiel für die Verknüpfung von Zugehörigkeitswerten

Als Spezialfall der Theorie unscharfer Mengen kann die Zugehörigkeit einer Aussage zur Menge der wahren Aussagen bzw. ihrer Komplementärmenge, der Menge der falschen Aussagen angesehen werden. Während bei der Booleschen Algebra eine Aussage immer entweder wahr oder falsch ist, erhält jede Aussage nach der Theorie der unscharfen Mengen einen "Wahrheitswert", der zwischen 0 (falsch) und 1 (wahr) liegt[1]. Auf diese Weise lassen sich Wahrheitswerte von Aussagen nach den oben genannten Rechenregeln verknüpfen.

---

1) Vgl. Negoita (59), S.82 ff.

2.4 Hilfsmittel zur Konstruktion von Expertensystemen

2.4.1 Vorgefertigte Shells

Es wurde bereits darauf hingewiesen, daß eine Expertensystem-Shell, die zusammen mit einer Wissensbasis für ein bestimmtes Fachgebiet konstruiert wurde, prinzipiell auch für jedes andere Fachgebiet verwendet werden kann. Diesem Gedanken folgend liegt es nahe, eine Shell aus einem bereits existierenden und erprobten Expertensystem für das eigene Problem zu verwenden.

Tatsächlich ist dieses Prinzip bereits erfolgreich angewandt worden[1], und es sind kommerzielle Shells für die unterschiedlichsten Computersysteme erhältlich[2]. Das Verwenden einer vorgefertigten Shell bringt eine Reihe von wichtigen Vorteilen:

- die Entwicklungszeit für ein Expertensystem kann erheblich gesenkt werden, da die Zeit für das Konstruieren und Testen der Shell entfällt;

- in der Regel sind für die Konstruktion der Wissensbasis keine speziellen Programmierkenntnisse erforderlich, d.h. es werden geringere Ansprüche an die softwaretechnische Ausbildung der Konstrukteure gestellt;

- es existiert meist wenigstens ein fertiges, in der Praxis erprobtes Expertensystem, das als Modell und Maßstab für das eigene System dienen kann.

---

1) Vgl. etwa das Beispiel EMYCIN bei Jackson (39), S.107 ff.
2) Vgl. den Katalog bei Waterman (91), S.339 ff.

Diesen Vorteilen stehen jedoch auch wichtige Nachteile gegenüber:

vorgefertigte Shells sind häufig stark auf die ursprüngliche Anwendung zugeschnitten, d.h. sie sind inflexibel in Bezug auf die zugrundeliegende Problemstruktur;

fertige Shells verursachen in der Regel erhebliche Anschaffungsausgaben und stellen oft hohe Ansprüche an die vorhandene Hardware und Systemsoftware;

es besteht die Gefahr von Fehlern in der Wissensbasis, die dadurch zustande kommen, daß das Expertensystem von Fachgebiets-Experten ohne Kenntnis der Eigenheiten und Beschränkungen der Expertensystem-Technologie erstellt wird bzw. die zugrundeliegenden Methoden nicht voll verstanden werden. Aus diesem Grund ist in jedem Fall das Hinzuziehen eines erfahrenen Knowledge-Engineers[1] ratsam, auch wenn die Shell selbst von Laien problemlos bedient werden kann.

Das Problem der Inflexibilität resultiert aus der Tatsache, daß es verschiedene Problemkategorien gibt, die unabhängig von dem jeweiligen Fachgebiet unterschiedliche Anforderungen an die Shell stellen. Diese Anforderungen können sich z.B. auf die Benutzeroberfläche (etwa die Möglichkeit zur graphischen Darstellung), die Wissensrepräsentation[2], die zugrundeliegenden Problemlösungsstrategien[3] oder die Verarbeitungsgeschwindigkeit beziehen.

---

1) Vgl. Abschnitt 3.1.
2) Vgl. Abschnitt 2.3.
3) Vgl. Puppe (68), S.3 ff. sowie Abschnitt 3.3.1.

So mag z.B. eine Shell, die zur Konstruktion eines Expertensystems im Bereich der medizinischen Diagnose entwickelt wurde, auch im Bereich der Fehlerdiagnose in Computern oder der Interpretation von physikalischen Meßergebnissen erfolgreich eingesetzt werden. Sie wird dagegen kaum geeignet sein, um ein Expertensystem zur Produktionsplanung oder zur Echtzeit-Steuerung chemischer Prozesse zu erstellen, da sie vermutlich weder über geeignete Methoden zur Erstellung von Plänen noch eine ausreichende Verarbeitungsgeschwindigkeit verfügt.

Das Problem der Inflexibilität wird jedoch zunehmend dadurch überwunden, daß unterschiedliche Problemlösungs- bzw. Wissensrepräsentations-Methoden in einer Shell implementiert werden[1]. In Abschnitt 3.3.1 wird versucht, einen überblick über die verschiedenen Problemkategorien und Problemlösungs-Strategien zu geben.

Insgesamt läßt sich feststellen, daß vorgefertigte Shells gut geeignet sind, um möglichst schnell Prototyp-Systeme zu erstellen und so die Anwendungsmöglichkeiten für Expertensysteme in einem Problembereich zu untersuchen, sofern gesichert ist, daß die Shell die für die Bewältigung der zugrundeliegenden Problemstruktur erforderlichen Eigenschaften aufweist.

Für das Anliegen dieser Arbeit, nämlich die Anwendungsmöglichkeiten der Expertensystem-Technologie im Bereich der Betriebswirtschaft zu untersuchen, wäre eine vorgefertigte Shell somit das geeignete Mittel gewesen. Dem Verfasser standen jedoch weder eine Shell noch die Mittel zu ihrer Beschaffung zur Verfügung. Zudem konnten durch die Konstruktion einer eigenen Shell wertvolle Erfahrungen im Umgang mit der zugrundeliegenden Technologie gesammelt werden.

---

1) Vgl. di Primio, Brewka (67), S.33.

2.4.2 Knowledge-Engineering-Sprachen

Unter Knowledge-Engineering-Sprachen versteht man Programmiersprachen, die auf einem sehr hohen Niveau die Konstruktion von Expertensystemen ermöglichen und speziell für diesen Zweck geschaffen wurden[1]. Das Niveau dieser Sprachen läßt sich etwa mit dem spezieller Programmiersprachen zur Statistik[2] oder Simulation[3] vergleichen.

Merkmal dieser Sprachen ist, daß die wesentlichen, problemunabhängigen Elemente des Expertensystems, d.h. die einzelnen Komponenten der Shell, bereits feste Bestandteile des Sprachsystems sind. So läßt sich etwa mit wenigen Befehlen eine komplette Benutzerschnittstelle oder Erklärungskomponente definieren. Der Konstrukteur kann sich somit auf die Formulierung des Wissens konzentrieren, die in der Regel durch entsprechend vorgefertigte Datenstrukturen erleichtert wird.

Knowledge-Engineering-Sprachen sind eine Mischung aus vorgefertigter Shell und allgemeiner Programmiersprache. Sie sind flexibler und vielseitiger als die Shell, sind jedoch zum Teil ebenfalls auf bestimmte Wissensrepräsentationsformen festgelegt. Sie sind entweder als Spracherweiterung bereits vorhandener allgemeiner Programmiersprachen[4] oder als eigenständige Sprachen[5] implementiert.

1) Vgl. Waterman (91), S.80.
2) Z.B. SPSS (Statistical Package for the Social Sciences); vgl. Beutel, Schubö (5).
3) Z.B. GPSS (General Purpose Simulation System); vgl. Bobilier, Kahan, Probst (7).
4) Z.B. LOOPS, FLAVORS; vgl. Waterman (91), S.352 f.
5) Z.B. ROSIE, OPS5; vgl. Waterman (91), S.118 ff.

Auch bei der Verwendung einer Knowledge-Engineering-Sprache gilt, daß sie zu der gegebenen Problemstruktur "passen" muß. Von Vorteil ist es, wenn es möglich ist, Programmteile in der zugrundeliegenden allgemeinen Programmiersprache in das Expertensystem zu integrieren, wie dies bei Spracherweiterungen immer der Fall ist. Auf diese Weise erhält man eine wesentlich vergrößerte Flexibilität.

2.4.3 Allgemeine Programmiersprachen

Unter einer allgemeinen Programmiersprache soll hier eine Programmiersprache verstanden werden, die nicht für einen speziellen Anwendungszweck entworfen wurde. Es ist allgemein möglich, ein beliebiges Problem, sofern es überhaupt programmtechnisch lösbar ist, in jeder dieser Programmiersprachen zu lösen. Allerdings eignen sich diese für verschiedene Problemtypen unterschiedlich gut. Einige typische Vertreter der allgemeinen Programmiersprachen sind FORTRAN, PASCAL und BASIC.

Im Bereich der künstlichen Intelligenz werden einige allgemeine Programmiersprachen verwendet, die zwar prinzipiell für jedes beliebige Problem einsetzbar sind, die sich jedoch aufgrund ihrer sehr flexiblen Datenstrukturen besonders für Anwendungen in diesem Bereich eignen. Die wichtigsten sollen im folgenden stichwortartig beschrieben werden.

Die älteste Programmiersprache, die in der künstlichen Intelligenz sehr weitgehende Verbreitung findet, ist LISP (List Processing Language)[1].

---

1) Eine fundierte Einführung in das Programmieren in LISP geben z.B. Winston, Horn (94).

Wie der Name andeutet, ist die grundlegende und einzige Datenstruktur der Sprache die Liste. Eine Liste ist definiert als eine geordnete Menge von Elementen, die entweder Atome (Symbole bzw. Zahlen) oder ebenfalls Listen sein können. Auch die LISP-Programme selbst haben die Form von Listen, so daß kein formaler Unterschied zwischen Daten und Programmteilen besteht. Dies gibt der Sprache eine außerordentliche Flexibilität und die Möglichkeit, sehr leicht sich selbst modifizierende Programme zu schreiben.

Der größte Nachteil von LISP ist die ausgesprochen unübersichtliche Listenschreibweise. In einzelnen Programmteilen können leicht Listengebilde mit einer zweistelligen Zahl von Klammerebenen entstehen, die vor allem unerfahrenen Programmierern das Verstehen eines Programms und die Fehlersuche erschweren. LISP wird insbesondere in den Vereinigten Staaten als grundlegende Sprache für nahezu alle Anwendungen der künstlichen Intelligenz benutzt. Es gibt eine Reihe von Spracherweiterungen, die das Programmieren von Expertensystemen erleichtern sollen[1].

In Europa und Japan wird verstärkt der Einsatz von PROLOG (Programming in Logic) für Anwendungen der künstlichen Intelligenz eingesetzt. Diese Sprache wurde von Kowalski entwickelt und von Colmerauer 1972 erstmals anwendungsreif implementiert[2]. Sie wird häufig als "deklarative Programmiersprache" bezeichnet[3].

---

1) Vgl. Waterman (91), S.339 ff.
2) Vgl. Clocksin, Mellish (14), S.V; Giannesini u.a. (29), S.V; von Zelewski (99), S.34.
3) Vgl. z.B. Harmon, King (33), S.103.

"Deklarativ" bedeutet dabei, daß in einem Programm nur noch spezifiziert wird, was zu tun ist, und nicht mehr, wie dies bei prozeduralen Sprachen erforderlich ist, wie es getan werden soll. Das "Wie" wird in PROLOG durch einen eingebauten Inferenzmechanismus selbst gelöst, der nach dem Prinzip der Rückwärtsverkettung alle möglichen Lösungen für eine gestellte Aufgabe sucht[1].

In der Praxis ist PROLOG jedoch von dem Ideal einer deklarativen Programmiersprache noch weit entfernt, da es eindeutig Eigenschaften einer prozeduralen Programmiersprache[2] aufweist. Grundlegendes Sprachelement von PROLOG ist die sog. Horn Klausel[3], die formal der Produktionsregel stark ähnelt. Die Sprache enthält auch "Metasprachliche Elemente", d.h. Programme können sich wie in LISP selbst verändern. Die Liste ist eine grundlegende, jedoch nicht die einzige Datenstruktur in PROLOG.

Durch den eingebauten Inferenzmechanismus eignet sich PROLOG sehr gut zur Konstruktion von Expertensystemen. Gegenüber LISP zeichnet sich die Sprache durch eine bessere Verständlichkeit des Programmtextes aus. Zudem verfügt PROLOG über leistungsfähige Sprachelemente zur Datenverwaltung. Nachteilig ist, daß PROLOG-Programme zu Ineffizienz und langer Ausführungsdauer neigen[4].

---

1) Vgl. Abschnitt 2.2.2.
2) Z.B. den sog. "Cut"; vgl. Clocksin, Mellish (14), S.58 ff.
3) Vgl. ebenda, S.222.
4) Vgl. Waterman (91), S.100.

Die dritte wichtige Programmiersprache für Anwendungen im Bereich der Künstlichen Intelligenz ist SMALLTALK[1]. Diese Sprache verwirklicht das Prinzip des objektorientierten Programmierens bereits auf der Ebene des Betriebssystems. Da die Sprache noch sehr jung ist, liegen noch kaum praktische Erfahrungen im Bereich der Konstruktion von Expertensystemen vor.

Der Verfasser hatte bisher nicht die Gelegenheit, Programmiererfahrungen in SMALLTALK zu sammeln, so daß an dieser Stelle relativ wenig über die Anwendbarkeit dieser Sprache gesagt werden kann. Wesentliche Stärken der Sprache scheinen in ihrer sehr hohen Modularität und der Gestaltung der Benutzeroberfläche zu liegen.

Zusammenfassend kann gesagt werden, daß alle hier genannten Programmiersprachen für die Entwicklung von Expertensystemen prinzipiell geeignet sind. Allerdings ist der Entwicklungsaufwand gegenüber den Knowledge-Engineering-Sprachen oder vorgefertigten Shells wesentlich größer, da die einzelnen Komponenten der Shell erst noch erstellt werden müssen. Dem steht jedoch der Vorteil größtmöglicher Flexibilität gegenüber.

---

1) Vgl. Geller (28), S.1049 sowie Abschnitt 2.3.4.

Natürlich ist es auch möglich, Expertensysteme in den "klassischen" Programmiersprachen wie PASCAL oder gar BASIC zu implementieren. Dies erfordert jedoch einen sehr hohen Entwicklungsaufwand, und Änderungen in der Struktur der Wissensbasis oder einzelnen Komponenten der Shell sind meist sehr mühselig. Der Vorteil der klassischen Programmiersprachen liegt in der oft grösseren Ausführungsgeschwindigkeit und dem geringeren Speicherplatzbedarf. Aus diesem Grunde sind Expertensysteme teilweise in einer der oben genannten Sprachen entwickelt und dann für den praktischen Einsatz in eine klassische Sprache übertragen worden[1].

---

1) Vgl. Waterman (91), S.150.

2.5 Die Expertensystem-Shell MOSES

2.5.1 Zielsetzung und Implementierung

Entsprechend der in Abschnitt 1.1 angeführten Zielsetzung wurde im Rahmen der vorliegenden Arbeit die "Modulare Shell für Expertensysteme" (MOSES) entwickelt. Dabei wurde einerseits das Ziel verfolgt, durch die praktische Anwendung der Programmiertechniken der Künstlichen Intelligenz fundierte Kenntnisse und Erfahrungen auf diesem Gebiet zu sammeln. Zum anderen konnten bei der Entwicklung die speziellen Anforderungen betriebswirtschaftlicher Fragestellungen, wie etwa die Notwendigkeit zur Verarbeitung von Zahlen, berücksichtigt werden. Schließlich sollte durch die Entwicklung einer Shell die Basis für weitere wissenschaftliche Arbeiten auf dem Gebiet betriebswirtschaftlicher Expertensysteme geschaffen werden.

Nach Abschluß des Projektes kann gesagt werden, daß tatsächlich tiefgehende Einblicke und Erfahrungen im Umgang mit den Methoden der künstlichen Intelligenz gesammelt werden konnten und sich der hohe Arbeitsaufwand für das Erlernen der notwendigen Programmiertechniken und das Entwickeln der Shell gelohnt hat.

Zu Beginn mußten die Anforderungen an eine derartige Shell definiert werden:

Die Shell sollte auch über den Rahmen dieser Arbeit hinaus zu Forschungszwecken einsetzbar sein. Sie sollte deshalb leicht bedienbar und flexibel genug sein, um für unterschiedliche Problembereiche anwendbar zu sein.

Da bei betriebswirtschaftlichen Analysen sehr häufig Zahlen betrachtet und verglichen werden müssen, sollte die Verarbeitung von numerischen Werten durch Einbindung entsprechender Unterprogramme möglich sein.

- Ebenso sollte es möglich sein, auf vorhandene Datenbestände durch entsprechende Unterprogramme zuzugreifen, um eventuell die Datenerfassung zu erleichtern.

- Der Inferenzmechanismus sollte in der Lage sein, auch unsicheres und unscharfes Wissen zu verarbeiten.

- Die Shell sollte alle wesentlichen Elemente eines Expertensystems, insbesondere eine aussagefähige Erklärungskomponente enthalten.

Das resultierende System MOSES wurde auf einem IBM-PC-kompatiblen Rechner in der Sprache TURBO PROLOG[1] implementiert. Das Listing ist in Anhang B wiedergegeben. Für den Einsatz der Shell sind ein Hauptspeicher von mindestens 512 KByte sowie zwei Diskettenlaufwerke oder eine Festplatte notwendig. Die Entwicklungszeit inklusive der Tests betrug etwa 5 Mannmonate.

Für TURBO PROLOG als Entwicklungswerkzeug sprach, daß dieses als Compiler konzipierte System einen sehr schnellen Programmcode erzeugt und eine komfortable Entwicklungsumgebung bietet. Zudem verfügt es über Sprachelemente, die die Gestaltung einer komfortablen Benutzeroberfläche des Expertensystems sehr erleichtern, wie z.B. einen integrierten Text-Editor und komfortable Fenstertechnik[2]. Allerdings weist TURBO PROLOG einige Einschränkungen gegenüber dem sog. Edinburgh-Standard für PROLOG[3] auf, was die Implementierung des Inferenzmechanismus etwas verkomplizierte.

1) TURBO PROLOG ist ein Produkt der Firma Borland International Inc. Deutscher Vertrieb: Heimsoeth Software, München.
2) Vgl. Heimsoeth (34), S.5 ff.
3) Vgl. Clocksin, Mellish (14), S.247 ff.

Aufgrund der genannten Anforderungen ergab sich die Notwendigkeit, TURBO PROLOG-Unterprogramme abhängig von der jeweiligen Problemstellung einzubinden. Dies führte zu einem modularen Aufbau der Shell, der es erlaubt, zusätzlich zu den Grundelementen Programmteile zu laden, die spezielle Aufgaben übernehmen können. Auf diese Weise ist es möglich, verschiedene Module, z.B. zur statistischen Analyse von Zahlenreihen oder zum Zugriff auf vorhandene Daten, zu erstellen und für eine gegebene Anwendung beliebig zu kombinieren.

2.5.2 Wissensdarstellung

Als Methode zur Wissensdarstellung wurde der Produktionsregel-Ansatz gewählt. Zum einen entspricht diese Methode der grundlegenden Struktur der Programmiersprache PROLOG, zum anderen sind die resultierenden Wissensbasen auch für Laien leicht nachvollziehbar und somit für Demonstrationssysteme besonders geeignet[1].

Da TURBO PROLOG nur unzureichende metasprachliche Elemente (d.h. die Möglichkeit, sich selbst modifizierende Programme zu schreiben) enthält, war es nicht möglich, das Wissen direkt als PROLOG-Regeln zu speichern. Stattdessen mußten eine eigene Regelform sowie ein entsprechender Regelinterpreter entwickelt werden.

Grundelement einer MOSES-Wissensbasis ist die sog. Aussage. Sie besteht aus drei Symbolen[2], die Zeichenketten beliebigen Inhalts sein können. Das erste und

1) Vgl. Abschnitt 2.3.1.
2) Mit "Symbol" ist in TURBO PROLOG eine nichtnumerische Konstante, d.h. eine alphanumerische Zeichenkette, gemeint; vgl. Heimsoeth (34), S.45.

dritte Symbol werden "Objekte", das mittlere "Relation" genannt. Diese Unterscheidung bezieht sich nicht auf den Inhalt der Symbole, sondern dient lediglich zur Identifikation des "Typs" einer Aussage: Aussagen mit gleicher Relation gelten als Aussagen des gleichen Typs. Der Typ einer Aussage wird zur Kommunikation des Systems mit PROLOG-Unterprogrammen benötigt.

Beispiele für zulässige MOSES-Aussagen sind:

Fahrzeug hat vier_Räder
Umsatz ist_grösser_als Kosten
A B C

Dabei dient der Unterstrich zum Verbinden mehrerer Worte zu einer Zeichenkette bzw. einem Symbol. Man erkennt, daß es keinerlei Beschränkungen des Inhalts von MOSES-Aussagen gibt.

Eine in Expertensystemen sehr oft verwendete Daten struktur ist das sog. Objekt-Attribut-Wert-Tripel[1]. Diese Struktur kann als Untermenge der MOSES-Aussagen angesehen werden, da eine Aussage als Objekt-Attribut Wert-Tripel interpretiert werden kann, wenn das linke Symbol als Objekt, das mittlere (die Relation) als At tribut und das rechte als Wert angesehen wird. Aller dings wird der "Wert" stets als Symbol interpretiert. Zahlenwerte können nur durch entsprechende PROLOG-Un- terprogramme verarbeitet werden[2].

Beispiele für Objekt-Attribut-Wert-Tripel sind:

Fahrzeug Farbe_ist gelb
Umsatz Höhe_ist 10000

---

1) Vgl. Harmon, King (33), S.45 ff.
2) Vgl. Abschnitt 2.5.6.

Abb. 2.6 zeigt beispielhaft eine MOSES-Regel in der von der Wissensakquisitions-Komponente verwendeten, annähernd natürlichsprachlichen Form sowie zum Vergleich in der vom Inferenzmechanismus verwendeten internen Form. Die Wissensakquisitionskomponente übersetzt eine eingegebene Regel in die interne Form, um eine effizientere Verarbeitung zu ermöglichen.

```
Regel Nr.5:
Gründung ist erfolgversprechend
   wenn Markt ist erfolgversprechend
   und Unternehmenskonzept ist erfolgversprechend
   und Finanzierung ist gesichert  (1)

rule (5, dann ("Gründung", "ist",
"erfolgversprechend"), (wenn ("Markt", "ist",
"erfolgversprechend"), wenn ("Unternehmenskonzept",
"ist", "erfolgversprechend"), wenn ("Finanzierung",
"ist", "gesichert")), 1)
```

Abb. 2.6: MOSES-Regel in externer und interner Darstellung

Eine MOSES-Regel besteht im wesentlichen aus drei Elementen: einem sog. Regelkopf, der den Folgerungsteil (DANN-Teil) in Form einer Aussage enthält; einem Regelkörper, der den Bedingungsteil (WENN-Teil) in Form einer geordneten Menge von Aussagen enthält; und einem sog. Wahrheitswert (Glaubwürdigkeitswert), der angibt, ob eine Regel immer gilt oder auch Ausnahmen denkbar sind. Ein Glaubwürdigkeitswert von 1 würde bedeuten, daß die Regel immer zutrifft, während bei einem Wert von z.B. 0,95 einige relativ seltene Ausnahmen denkbar wären. Ein Wert von 0 würde bedeuten, daß die Regel nie zuträfe; eine derartige Regel wäre folglich sinnlos.

Die Aussage des Regelkopfes wird als wahr angesehen, wenn alle Aussagen des Regelkörpers wahr sind ("Und-Verknüpfung" bzw. Konjunktion). Dabei kann im Regelkörper auch die Negation einer Aussage gefordert sein, indem der Aussage das Wort "nicht" vorangestellt wird. Der Wahrheitswert der Negation ist dann die Differenz zwischen 1 und dem Wahrheitswert der negierten Aussage.

Ein weiteres Element der Wissensbasis sind die sog. Fakten. Dies sind Aussagen, die mit einem Wahrheitswert versehen sind. Innerhalb einer MOSES-Wissensbasis werden Fakten als Wissen über ein konkretes Problem interpretiert, während Regeln als allgemeingültiges Wissen über das jeweilige Fachgebiet angesehen werden.

Fakten werden im allgemeinen durch Befragung des Benutzers erlangt. Dies geschieht, indem ein vordefinierter Fragetext für die entsprechende Aussage ausgegeben wird. Der Benutzer kann dann die Frage durch Eingabe eines Wahrheitswertes beantworten. Die Aussage wird zusammen mit dem angegebenen Wahrheitswert als Faktum gespeichert.

Es ist auch möglich, Fakten mit Hilfe der Wissensakquisitionkomponente ohne vorherige Frage des Systems einzugeben. Dabei muß allerdings auf eine mit der entsprechenden Aussage in den Regeln übereinstimmende Schreibweise geachtet werden, da das Faktum sonst nicht erkannt wird. Aus diesem Grunde eignet sich diese Form kaum für die Eingabe von Fakten durch den Endbenutzer. Um eine neue Problemstellung zu bearbeiten, können die Fakten über das alte Problem gelöscht oder auch geändert werden.

Hinweistexte schließlich dienen der Ausgabe von Informationen für den Benutzer. Jedem Hinweistext ist eine Aussage sowie eine Ober- und Untergrenze für den Wahrheitswert dieser Aussage zugeordnet. Liegt der Wahr-

heitswert der Aussage innerhalb dieser Grenzen, so wird der Hinweistext ausgegeben. Auf diese Weise können dem Benutzer Hinweise abhängig davon gegeben werden, ob die Aussage bestätigt werden kann oder nicht. Hinweistexte sind ein wichtiges Mittel, um die Ergebnisse einer Konsultation für den Benutzer verständlich darzustellen oder ihn auf bestimmte Zusammenhänge aufmerksam zu machen, die sich aus den von ihm angegebenen Fakten ergeben.

2.5.3 Die Inferenzkomponente

Abb. 2.7 zeigt die Arbeitsweise der Inferenzkomponente in Form eines Flußdiagramms. Die Befragung des Expertensystems wird gestartet, indem der Benutzer eine Aussage angibt, deren Wahrheitswert bestimmt werden soll. Diese Zielaussage kann auch bei der Erstellung der Wissensbasis vordefiniert werden, um das Starten durch den Benutzer zu vereinfachen.

Im ersten Schritt prüft MOSES, ob für den Typ der Aussage (d.h. ihre Relation) ein PROLOG-Unterprogramm in Form eines Moduls existiert. Ist dies der Fall, so wird die Aussage an das Unterprogramm übergeben, das einen Wahrheitswert als Ergebnis übermittelt. Auf diese Weise lassen sich PROLOG-Programmteile an jeder Stelle der Wissensbasis einfügen.

Existiert kein Unterprogramm, so versucht MOSES zunächst, ein Faktum für die Aussage zu finden. Ist dies der Fall, so wird der Aussage der Wahrheitswert des Faktums zugeordnet.

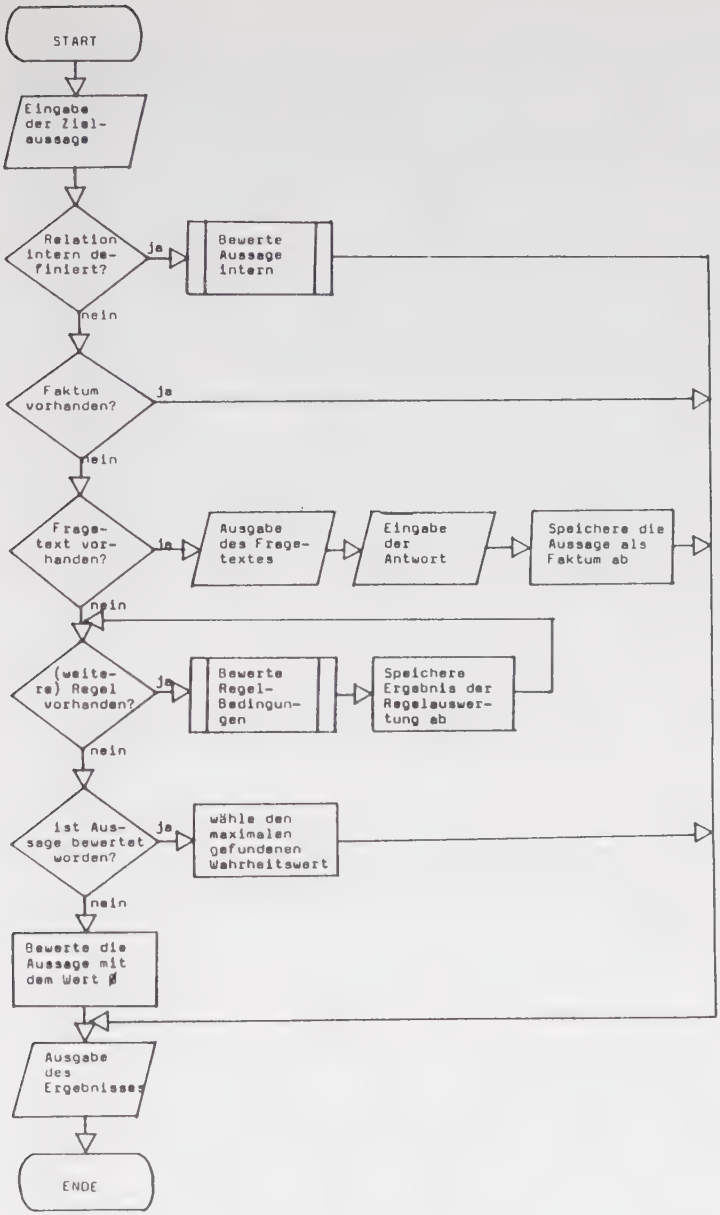


Abb. 2.7: Flußdiagramm der MOSES-Inferenzkomponente

Wird für die Aussage kein Faktum gefunden, so wird nach einem Fragetext für diese Aussage gesucht und dieser ggf. an den Benutzer ausgegeben. Die Aussage wird dann mit dem vom Benutzer geantworteten Wahrheitswert bewertet und als Faktum abgespeichert, um späteres Stellen derselben Frage zu vermeiden.

Existiert für die Zielaussage weder ein Faktum noch ein Fragetext, so wird nach Regeln gesucht, deren Regelkopf die Zielaussage enthält. Wird eine solche gefunden, so wird versucht, die einzelnen Aussagen im Regelkörper zu bewerten. Dazu werden diese ebenfalls zu Zielaussagen erhoben und der Schlußfolgerungsmechanismus rekursiv auf sie angewandt. Sind auf diese Weise alle Bedingungen bewertet worden, wird der minimale Wahrheitswert dieser Aussagen, multipliziert mit dem Glaubwürdigkeitswert der Regel, der ursprünglichen Zielaussage als Wahrheitswert zugeordnet, und es wird versucht, durch Anwendung weiterer Regeln einen höheren Wahrheitswert zu finden. Existieren weder Regeln, Fragen noch ein Fragetext für die Aussage, wird ein Wahrheitswert von 0 unterstellt.

Die Arbeitsweise der MOSES-Inferenzkomponente soll an einem Beispiel verdeutlicht werden. Es basiert auf der Wissensbasis des Expertensystems FOUNDER, die in Anhang A wiedergegeben ist.

Gegeben sei die Zielaussage "Gründer ist körperlich_belastbar". Das System prüft zunächst, ob die Relation "ist" intern definiert ist, d.h. ob diese Aussage durch ein spezielles Unterprogramm bewertet werden soll. Dies ist nicht der Fall, daher prüft das System nun, ob die Aussage bereits als Faktum in der Wissensbasis enthalten ist. Da auch dies nicht der Fall ist, wird nach einem Fragetext für diese Aussage gesucht, der jedoch ebenfalls nicht existiert. Schließlich sucht das System

nach einer Regel, die die Aussage "Gründer ist körperlich_belastbar" im Regelkopf enthält, und findet

Regel Nr.13 (Glaubwürdigkeit 1) :
Gründer ist körperlich_belastbar
  wenn Gründer treibt regelmäßig_Sport

Der Regelkörper dieser Regel enthält nur eine Bedingung, nämlich die Aussage "Gründer treibt regelmäßig_Sport". Für diese Aussage existiert der Fragetext

"Treiben Sie regelmäßig (mindestens einmal in der Woche) Sport?"

der an den Benutzer ausgegeben wird. Antwortet dieser beispielsweise mit "nein", so wird diese Aussage mit dem Wahrheitswert 0 als Faktum abgespeichert. Die Aussage "Gründer ist körperlich_belastbar" wird nun vorläufig mit dem Wert 0 bewertet. Die Inferenzkomponente sucht nach weiteren Regeln für die Zielaussage und findet

Regel Nr.14 (Glaubwürdigkeit 0,8) :
Gründer ist körperlich_belastbar
  wenn nicht Gründer ist krank
  und Gründer fühlt_sich fit

Die nächste Zielaussage ist daher "Gründer ist krank". Ein entsprechender Fragetext wird nun ausgegeben, den der Benutzer mit "wahrscheinlich nicht" (Wahrheitswert 0,2) beantwortet. Der Wahrheitswert der ersten Bedingung ist damit, da die Negation dieser Aussage gefordert wird, 1 - 0,2 = 0,8.

Die zweite Bedingung lautet "Gründer fühlt_sich fit". Auch hierfür existiert ein Fragetext, den der Benutzer mit "ja" (Wahrheitswert 1) beantwortet. Der Wahrheitswert der Aussage im Regelkopf ergibt sich nun als Minimum der beiden Wahrheitswerte der Bedingungen multipliziert mit dem Glaubwürdigkeitswert der Regel:

min (0,8; 1) * 0,8 = 0,64.

Die Zielaussage "Gründer ist körperlich_belastbar" war bereits durch Anwendung von Regel Nr.13 mit 0 bewertet worden. Da keine weiteren Regeln für diese Aussage existieren, ergibt sich der Wahrheitswert der Zielaussage als Maximum der durch die einzelnen Regeln erzielten Bewertungen:

max (0; 0,64) = 0,64.
====

2.5.4 Die Dialogkomponente

Die Dialogkomponente hat die Aufgabe, die Kommunikation des Benutzers mit dem Expertensystem zu steuern[1]. Dies umfaßt das Starten der Befragung, die Ausgabe von Fragen, die Eingabe der Antworten und die Ausgabe von Hinweistexten.

Wenn das MOSES-System gestartet wird, fragt es den Benutzer zunächst nach dem Namen der Wissensbasis. Hier zeigt sich bereits deutlich die klare Trennung zwischen problemabhängigen und problemunabhängigen Bestandteilen des Expertensystems. Die angegebene Wissensbasis wird dann geladen, und es kann mit der Befragung begonnen werden.

---

1) Vgl. Abschnitt 2.2.4

Existiert die spezifizierte Wissensbasis noch nicht, so startet MOSES als "leeres System", d.h. es ist noch kein Wissen vorhanden. Eine neue Wissensbasis kann dann mit Hilfe der Wissensakquisitionskomponente erstellt werden.

Abb. 2.8 zeigt den Bildschirminhalt zu Beginn einer Befragung. MOSES benutzt für die Kommunikation mit dem Benutzer zwei voneinander unabhängige Bildschirmfenster. Die obere Bildschirmhälfte bildet das Dialogfenster. Hier kann der Benutzer eine Zielaussage vorgeben oder eine Funktion der Wissensakquisitions- oder Erklärungskomponente wählen. Fragen des Systems werden ebenfalls im Dialogfenster ausgegeben.

```
------------------------------ * MOSES V 1.0 * ------------------------------
------------------------------- founder -------------------------------------
 * Wie-Trace ausgeschaltet.

 * Freier Speicherplatz: 304159 Bytes.

>

------------------------------------------------------------------------------
------------------------- Eingabemöglichkeiten -------------------------------
  * - Befragung beginnen

  ? - Eingabemöglichkeiten              w - Ein/Ausschalten des Wie-Trace
  r - Ändern von Regeln                 s - Speichern der Wissensbasis
  f - Ändern von Fakten                 b - Beenden der Konsultation
  l - Löschen aller Fakten              m - freier Speicherplatz
  i - Interne Relationen sehen          e - Erläuterungen sehen/ändern
  t - Fragetexte erstellen              h - Hinweise erstellen
  g - Grenz-Wahrheitswert festlegen     z - Ziel definieren
------------------------------------------------------------------------------
```

Abb. 2.8: MOSES-Bildschirmaufbau

Während das Dialogfenster immer dieselbe Gestalt und Funktion hat, überlagern sich auf der unteren Bildschirmhälfte eine Vielzahl von Fenstern. Es ist immer jeweils nur das Fenster sichtbar, das zuletzt vom System angesprochen wurde. Die untere Bildschirmhälfte wird vor allem von der Erklärungs- und Wissensakquisitionskomponente genutzt. Die Dialogkomponente gibt dort Hilfs- und Hinweistexte aus.

Eine MOSES-Befragung wird dadurch ausgelöst, daß eine Ziel-Aussage der Form "Objekt1 Relation Objekt2" eingegeben wird. Dabei können die Objekte durch Variablen der Form "X" oder "Y" ersetzt werden. In diesem Fall werden nacheinander alle Aussagen zu Zielaussagen, die in einem Regelkopf enthalten sind und die zu der vorgegebenen Zielaussage "passen". Dies ist gegeben, wenn die nicht durch Variablen ersetzten Bestandteile der Zielaussage mit denen der betrachteten Aussage übereinstimmen. Wird beispielsweise die Aussage "X ist defekt" eingegeben, so werden zunächst alle Regeln gesucht, deren Regelkopf die Form "... ist defekt" hat. Dann wird jeder dieser Regelköpfe zur Zielaussage.

Um den Start des Systems für den Benutzer zu erleichtern, kann eine Zielaussage vordefiniert werden. Die Befragung kann dann durch Eingabe des Stern-Symbols ("*") ausgelöst werden.

Wird ein Fragetext an den Benutzer ausgegeben, so kann er diesen entweder durch einen Zahlenwert zwischen 0 (nein) und 1 (ja) oder durch eine verbale Bewertung wie etwa "weiß nicht" oder "wahrscheinlich nicht" beantworten. Die verbale Bewertung wird von der Dialogkomponente anhand einer internen Tabelle in einen Wahrheitswert umgerechnet. So entspricht etwa der Antwort "wahrscheinlich nicht" ein Wahrheitswert von 0,2. Gibt der Benutzer statt einer Antwort ein Fragezeichen ein, so wird die Umrechnungstabelle im unteren Bildschirmfenster angezeigt.

2.5.5 Die Erklärungskomponente

MOSES unterscheidet zwei verschiedene Formen der Erklärung[1]: "Warum-Erklärungen" und einen sog. "Wie-Trace" (Trace = engl. Spur).

Warum-Erklärungen können jederzeit angefordert werden, wenn das System die Beantwortung einer Frage vom Benutzer erwartet. Durch Eingabe des Wortes "Warum" anstelle einer Antwort wird dann im unteren Fenster die zuletzt benutzte Regel angezeigt, die zur Stellung der Frage geführt hat. Der Benutzer kann nun entscheiden, ob er mit der Beantwortung der Frage fortfahren möchte oder ob er den sog. Erklärungspfad weiter verfolgen möchte. Abb. 2.9 zeigt einen Ausschnitt aus einer Befragung, bei der der Benutzer eine Warum-Erklärung verlangt.

```
------------------------------ " MOSES V 1.0 " ------------------------------
 ------------------------------- founder ------------------------------------
 )"

 Leiden Sie an chronischen Erkrankungen (wie z.B. Herzleiden), die Ihre
 Leistungsfähigkeit und Belastbarkeit beeinträchtigen? nein


 Fühlen Sie sich fit genug, um über einen längeren Zeitraum eine 60-Stunden-
 Woche durchzuhalten? warum

 ------------------------------------------------------------------------------
 ------------------------------- Warum ------------------------------------
 Ich versuche Regel 14 anzuwenden:
 Gründer ist körperlich_belastbar
    wenn nicht Gründer ist krank
    und Gründer fühlt_sich fit
 (W: 1)
 Weiter verfolgen (j/n)?
```

Abb. 2.9: Warum-Erklärung

1) Vgl. Abschnitt 2.2.4.

Der Erklärungspfad besteht aus einer Kette von Regeln, die von der gestellten Frage bis zur ursprünglichen Zielaussage führt. Möchte der Benutzer den Erklärungspfad weiter verfolgen, so wird jeweils diejenige Regel angezeigt, die ihrerseits zur Anwendung der zuvor gezeigten Regel geführt hat. Auf diese Weise läßt sich der Suchbaum bis zum Ursprung, d.h. bis zur ursprünglichen Zielaussage, zurückverfolgen. Der Erklärungsprozess kann an jeder beliebigen Stelle abgebrochen werden.

Warum-Erklärungen können zum einen dazu dienen, die Schlußfolgerungen des Systems zu verfolgen und somit die Richtigkeit der Wissensbasis zu prüfen. Zum anderen können sie dem Anwender den Grund für eine gestellte Frage zeigen und es ihm somit erleichtern, dem Kontext entsprechend zu antworten. Auf diese Weise wird auch die Akzeptanz des Systems gefördert. Außerdem kann die Warum-Erklärung dem Benutzer die Konsequenzen einer bestimmten Antwort vor Augen führen und somit Wissen an den Benutzer vermitteln.

Der sog. Wie-Trace liefert keine Erklärungen im eigentlichen Sinn. Vielmehr erlaubt er es, die Schlußfolgerungen des Systems während einer Konsultation zu beobachten. Dies geschieht, indem jeder Einzelschritt der Inferenzkomponente im unteren Fenster angezeigt wird. Die Befragung kann an jedem beliebigen Punkt durch Drücken der Escape-Taste abgebrochen werden. Der Wie-Trace kann vor einer Befragung an- oder ausgeschaltet werden. Da jede beliebige Aussage als Zielaussage angegeben werden kann, läßt sich jederzeit prüfen, wie das System zu einer bestimmten Schlußfolgerung oder zu einem Teilergebnis gelangt ist.

Wenn das System zum Beweis einer Aussage eine Regel anführt, so wird diese Regel zusammen mit dem Hinweis "Ich wende Regel Nr. ... an:" im unteren Fenster angezeigt. Findet die Inferenzkomponente ein Faktum zu der gesuchten Aussage, so wird dieses mit dem Hinweis "Ich weiß:" und der Angabe des entsprechenden Wahrheitswertes in Klammern angezeigt.

Eine Schlußfolgerung des Systems wird durch den Hinweis "Ich folgere:", gefolgt von der Aussage und dem gefolgerten Wahrheitswert, dargestellt. Findet das System keinen Hinweis für die Richtigkeit einer Aussage, so zeigt es dies durch den Hinweis "Ich unterstelle:", gefolgt von der Aussage und dem Wahrheitswert 0, an.

Existiert für eine gegebene Aussage ein PROLOG-Unterprogramm, so meldet MOSES "Ich bewerte die Aussage ... intern". Der Ablauf innerhalb des Unterprogramms kann von der Erklärungskomponente nicht verfolgt werden. Es gibt jedoch die Möglichkeit, die Trace-Funktion des TURBO PROLOG-Systems zu nutzen, um die Schlußfolgerung innerhalb des Unterprogramms zu analysieren. Diese Funktion arbeitet ähnlich wie der Wie-Trace, jedoch auf einem detaillierteren Niveau, das bei normalen Schlußfolgerungen zu für den Laien unverständlichen Erklärungen führen würde.

Abb. 2.10 zeigt einen Ausschnitt aus dem Wie-Trace einer beispielhaften Befragung.

```
Ich wende Regel Nr. 3 an:
persönliche_Voraussetzungen sind erfüllt
   wenn Gründer ist körperlich_belastbar
   und Gründer ist geistig_belastbar
   und Familie ist hilfsbereit
   und Gründer hat Fachkenntnisse
   und Gründer hat Kenntnisse_der_Betriebsführung (1)

Ich wende Regel Nr. 14 an:
Gründer ist körperlich_belastbar
   wenn nicht Gründer ist krank
   und Gründer fühlt_sich fit (1)

Leiden Sie an chronischen Erkrankungen (wie z.B.
Herzleiden), die ihre Leistungsfähigkeit und Belast-
barkeit beeinträchtigen? nein

Ich merke mir:
  Gründer ist krank (0)

Ich folgere:
  Gründer ist krank (0)

Fühlen Sie sich fit genug, um über längere Zeit
eine 60-Stunden-Woche durchzuhalten? vielleicht

Ich merke mir:
  Gründer fühlt_sich fit (0.5)

Ich folgere:
  Gründer fühlt_sich fit (0.5)

Ich folgere:
  Gründer ist körperlich_belastbar (0.5)
```

Abb. 2.10: Ausschnitt aus einer MOSES-Befragung mit aktiviertem Wie-Trace

2.5.6 Die Wissensakquisitions-Komponente

Eine MOSES-Wissensbasis enthält verschiedene Formen von Wissen. Neben dem eigentlichen Regelwissen gibt es die Fakten, die in der Regel aufgrund von Benutzer Antworten gebildet werden und somit fallspezifisches Wissen gegenüber dem allgemeinen Fachwissen enthalten. Außerdem gibt es Frage- und Hinweistexte sowie eine vordefinierte Zielaussage und den Grenz-Wahrheitswert, die ebenfalls zur Wissensbasis gezählt werden können.

Regeln werden im unteren Bildschirmteil angezeigt und geändert. Durch Benutzen der Cursor-Pfeiltasten kann durch die Wissensbasis "geblättert" werden; durch Drükken der Return-Taste kann die gerade angezeigte Regel mit Hilfe des komfortablen TURBO PROLOG-Editors[1] geändert werden. Wenn ein Regeltext geändert oder neu erstellt wurde, muß anschließend der Wahrheitswert oder Gültigkeitsgrad der Regel angegeben werden. Es ist auch möglich, durch Angabe der Regelnummer eine Regel direkt auszuwählen.

Fakten können auf die gleiche Art geändert und hinzugefügt werden wie Regeln. Durch Ändern eines Faktums ist es möglich, nach Ende einer Befragung den Einfluß der Beantwortung einer bestimmten Frage auf die Lösung des Systems zu untersuchen. So können ebenfalls leicht falsche Antworten korrigiert oder alternative Antworten angegeben werden. Das System kann jederzeit auf der Basis der veränderten Fakten neu gestartet werden. Aufgrund der Arbeitsweise der Inferenzkomponente werden in vorherigen Befragungen bereits beantwortete Fragen nicht erneut gestellt.

---

1) Der Editor des TURBO PROLOG-Programmiersystems kann auch von TURBO PROLOG-Programmen aus aufgerufen und genutzt werden. Vgl. Heimsoeth (34), S.202 und S.218 ff.

Soll eine Befragung, z.B. für einen neuen Anwendungsfall, von vorn gestartet werden, ist es möglich, alle bisherigen Fakten gleichzeitig zu löschen. Dies empfiehlt sich ebenfalls, wenn Änderungen an der Regel-Wissensbasis vorgenommen wurden, da Fakten und Regeln immer gemeinsam abgespeichert werden. Wenn die vorherigen Antworten erhalten bleiben sollen, kann die gesamte Wissensbasis jederzeit unter einem beliebigen Dateinamen abgespeichert werden.

Fragetexte können durch Eingabe der entsprechenden Aussage angezeigt und geändert werden. Durch Eingabe des Schlüsselwortes "list" werden nacheinander alle Fragetexte mit den zugehörigen Aussagen angezeigt. Das gleiche gilt für die Erstellung und Änderung der Hinweistexte, wobei hier zusätzlich je ein minimaler und maximaler Wahrheitswert angegeben werden muß, so daß für eine Aussage abhängig von ihrem Wahrheitswert verschiedene Hinweise gegeben werden können. Die Änderung des Grenzwahrheitswertes und der vorgegebenen Zielaussage erfolgt durch einfache Neueingabe.

2.5.7 Intern definierte Relationen

TURBO PROLOG-Unterprogramme für intern definierte Relationen werden nicht zur Wissensbasis im engeren Sinne gerechnet, obwohl es sich bei ihnen teilweise auch um problembereichsspezifische Methoden handelt. Sie können nur innerhalb des TURBO PROLOG-Programmiersystems erstellt oder geändert werden und werden beim Start des Systems automatisch zusammen mit der Shell geladen, d.h. sie sind unabhängig von der danach geladenen Wissensbasis. Von MOSES aus kann lediglich eingesehen werden, welche Relationen intern definiert sind.

Anhang C zeigt ein Beispiel für ein TURBO PROLOG-Unterprogramm, das die Relation "Liquidität_ist" intern definiert. Die Prädikate "internal" und "is_internal" dienen dazu, dem MOSES-System mitzuteilen, welche Relationen intern definiert sind. Die doppelte Deklaration ist aus bestimmten, hier nicht näher zu erläuternden programmtechnischen Gründen notwendig.

Das Prädikat "ist_wahr (Objekt1, Relation, Objekt2, Wert)" bildet die Schnittstelle zum MOSES-System. Die Inferenzkomponente ruft, sofern die Relation intern definiert ist, dieses Prädikat auf, wobei die Variablen Objekt1, Relation und Objekt2 an die entsprechenden Teile der zu bewertenden Aussage "gebunden" sind[1]. Das Prädikat "ist_wahr" ruft nun seinerseits weitere, im Unterprogramm definierte Prädikate auf, die die Aussage bewerten, und bindet schließlich die Variable Wert an das Ergebnis dieser Bewertung. Der Inhalt der Variablen Wert wird von der MOSES-Inferenzkomponente als Wahrheitswert für die Aussage interpretiert.

über die Möglichkeit des TURBO PROLOG-Systems, Programm-Module zu sog. "Projekten" zu kombinieren[2], kann eine Bibliothek von intern definierten Relationen aufgebaut werden, die je nach Bedarf beliebig kombiniert werden können. Vor Beginn einer Befragung müssen dann ggf. die notwendigen Programm-Module in das MOSES System eingebunden werden. Da dieser Prozeß nur mit Hilfe des TURBO PROLOG-Systems möglich und nicht unkompliziert ist, empfiehlt es sich, ggf. verschiedene Versionen der MOSES-Shell abzuspeichern, die jeweils die für ein bestimmtes Expertensystem erforderlichen Module enthalten.

---

1) "Binden" bedeutet in PROLOG, einer Variablen einen temporären Wert zuzuweisen. Vgl. Mescheder (55), S.96.
2) Vgl. Heimsoeth (34), S.190 ff.

2.5.8 Kritische Würdigung

Verschiedene Tests und die Erstellung des in Kapitel 4 beschriebenen Expertensystems zur Existenzgründung haben gezeigt, daß sich mit der hier vorgestellten Shell funktionsfähige Expertensysteme einfach und komfortabel erstellen und überprüfen lassen. Eine Anwendung auch auf andere Fachgebiete insbesondere der Betriebswirtschaftslehre scheint vor allem aufgrund der Fähigkeit des Systems, durch geeignete Unterprogramme Zahlen und bereits vorhandene Datenbestände zu verarbeiten, möglich. Dennoch gibt es selbstverständlich Ansatzpunkte für Verbesserungsmöglichkeiten des Systems.

Die Schlußfolgerungskomponente könnte durch effizientere Suchstrategien zielstrebiger und schneller gestaltet werden. So wäre es z.B. denkbar, sog. "Meta-Regeln" in die Wissensbasis einzubauen, die Wissen darüber enthalten, unter welchen Umständen welche Regeln angewandt werden können[1]. Dies führt vor allem bei sehr großen Wissensbasen zu einer erheblichen Beschleunigung des Schlußfolgerungsprozesses.

Die Funktionen der Erklärungskomponente könnten komfortabler gestaltet werden. So erscheint z.B. eine Erweiterung der Warum-Erklärung dahingehend, daß auch alle bisher erzielten Zwischenergebnisse angezeigt werden, sinnvoll. Weiterhin wäre es denkbar, zu den einzelnen Regeln zusätzlich Erklärungstexte zu erstellen, die vom Benutzer abgefragt werden können und ihm den Sinn einer Regel allgemeinverständlich erklären. Eine derartige Erweiterung wäre vor allem für Systeme, die der Wissensvermittlung dienen sollen, wünschenswert.

1) Vgl. Morgado (56), S.432.

Die Benutzung der Dialog- und Wissensakquisitions-Komponenten könnte durch die Verwendung von sog. "Pull-Down-Menüs"[1] übersichtlicher gestaltet werden. Bei der Dialogkomponente wäre zudem daran zu denken, anstelle einfacher Ja Nein-Aussagen (mit den entsprechenden Abstufungen) auch Multiple-Choice-Antworten zuzulassen, wobei aus einer Menge möglicher, sich gegenseitig ausschließender Antworten (z.B. verschiedene Rechtsformen oder Branchen) die passende ausgewählt und als Faktum gespeichert werden könnte.

Der verwendete Ansatz der Darstellung von Wissen in Regelform ist, wie in Abschnitt 2.3.1 gezeigt, nicht sehr effizient, wenn das Wissen über den Problembereich hierarchisch strukturiert ist. Aus diesem Grunde wäre es für einige Anwendungsbereiche wünschenswert, auch die Wissensdarstellung in Form von Frames zu ermöglichen. Dies würde jedoch eine Veränderung der grundlegenden Systemarchitektur und damit einen erheblichen Arbeitsaufwand erfordern. Auch die Erklärungs-, Dialog- und Wissensakquisitions-Komponenten müßten in diesem Fall neu gestaltet werden.

Etwas weniger gravierend wäre die Erweiterung der Wissensdarstellung um sog. "strukturierte Regeln"[2] Gegenüber den hier verwendeten einfachen Regeln enthalten strukturierte Regeln zusätzlich Aktivierungsbedingungen, d.h. Umstände, die (ähnlich den Meta Regeln) das Anwenden der Regel auslösen, sowie Ausnahmen, bei denen die Regel nicht angewandt werden kann. Durch derartige Regeln ließe sich vor allem die Zahl der nötigen Regeln senken und die Wissensbasis somit effizienter und übersichtlicher gestalten.

1) "Pull-Down Menüs" sind eine heute gangige Methode der Bildschirmsteuerung zur Auswahl von Programmfunktionen; vgl. z.B. Heimsoeth (34), S.11 ff.
2) Vgl. Puppe (68), S.4 und S.11 f.

3 Die Anwendung der Expertensystem-Technologie

3.1 Die Erstellung eines Expertensystems

3.1.1 Phasen der Erstellung eines Expertensystems

In der Literatur wird der Prozess der Konstruktion eines Expertensystems in fünf Stufen unterteilt, die durch Rückkopplung miteinander verknüpft sind. Abbildung 3.1 veranschaulicht diesen Prozeß[1]. Bei diesem Konzept wird die Konstruktion der problemunabhängigen Elemente des Expertensystems ausgeklammert, d.h. es wird lediglich die Erstellung der Wissensbasis bei bereits vorhandener Shell betrachtet. Dieser Prozeß wird auch als "Knowledge Engineering" bezeichnet[2].

Im ersten Schritt, der Identifikation, muß die zugrundeliegende Problemstruktur erarbeitet werden. Es muß geprüft werden, ob das Problem hinsichtlich seiner Komplexität und wirtschaftlichen Bedeutung für eine Expertensystem-Anwendung geeignet erscheint oder ggf. eine Einschränkung des Problembereichs erforderlich ist. Weiterhin muß geprüft werden, ob die materiellen und personellen Ressourcen, die für die Erstellung eines Expertensystems notwendig sind, vorhanden sind oder beschafft werden können. Diese Voraussetzungen werden in Abschnitt 3.3.2 detailliert behandelt.

Auf der Grundlage der im ersten Schritt erarbeiteten Problemstruktur werden im zweiten Schritt Konzepte zur Formalisierung des Wissens erstellt. Weiterhin wird eine grundlegende Struktur des relevanten Wissens, z.B. in Form einer hierarchischen Gliederung, erarbeitet, die die Basis für die folgenden Schritte bildet.

---

1) Vgl. z.B. Noelke (62), S.112 ff.
2) Vgl. ebenda, S.109.
3) Vgl. Abschnitt 3.3.1

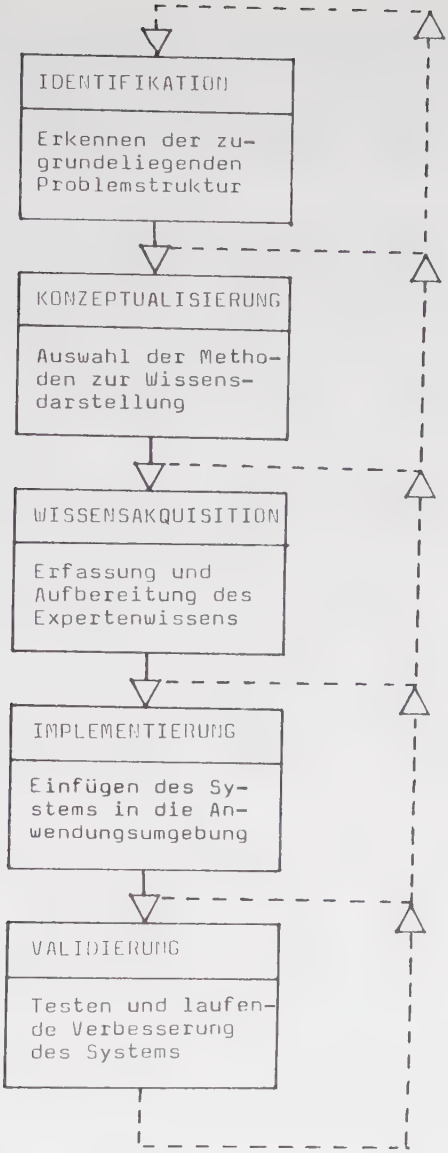


Abb. 3.1: Der Prozeß des "Knowledge Engineering"

Der dritte Schritt beinhaltet die Wissensakquisition. Dabei wird die Wissensbasis, meist außerhalb des späteren Anwendungsumfeldes, schrittweise erweitert und getestet. Dieser Prozeß wird im folgenden Abschnitt detailliert beschrieben.

Der vierte Schritt umfaßt die Implementierung des Systems im späteren Anwendungsumfeld. Dazu gehört ggf. die Umsetzung vom Entwicklungssystem auf die am Anwendungsort verfügbare Hardware sowie die Integration in das bereits vorhandene Informationssystem[1]. Weiterhin müssen die Anwender in den Gebrauch des Systems eingewiesen werden.

Im letzten Schritt muß die Leistungsfähigkeit des Expertensystems im praktischen Einsatz getestet werden. Dabei auftretende Mängel oder Schwächen müssen durch erneutes Durchlaufen einzelner Stufen - in der Regel Schritt 3 und 4 - verbessert werden. Schließlich muß das funktionsfähige System auch im praktischen Einsatz laufend erweitert und getestet werden, da sich das zur Lösung des Problems erforderliche Wissen in der Regel kontinuierlich ändert. Der Prozess der Wissensakquisition ist also praktisch niemals abgeschlossen.

3.1.2 Der Prozess der Wissensakquisition

Unter Wissensakquisition soll hier die schrittweise Erfassung, Aufbereitung, Kodierung und Verifizierung von Wissen für ein Expertensystem verstanden werden. Dabei wird davon ausgegangen, daß eine Shell und die grundlegenden Konzepte zur Wissensdarstellung bereits zuvor erarbeitet wurden.

---

1) Vgl. Abschnitt 3.4.2.

Die Wissensakquisition ist in der Regel der bedeutendste Zeit- und Kostenfaktor bei der Erstellung eines Expertensystems[1]. Die Sorgfalt bei der Akquisition des Wissens bestimmt die Qualität der Wissensbasis und damit die Leistungsfähigkeit des Expertensystems insgesamt[2].

Der Prozeß der Wissensakquisition läßt sich analog dem Schema im vorigen Abschnitt in vier Phasen unterteilen, die in Abb. 3.2 in einer übersicht dargestellt sind. Dabei werden die einzelnen Phasen so lange wiederholt, d.h. die Wissensbasis wird schrittweise vergrößert, bis das Expertensystem eine den Anforderungen der Praxis genügende "Kompetenz" erreicht hat. Erst danach wird das System in seine spätere Anwendungsumgebung integriert und dort im praktischen Alltag ausgiebig getestet und verbessert (Phasen 4 und 5 in Abb. 3.1).

Diese inkrementelle Vorgehensweise ist ein besonderes Kennzeichen des Knowledge Engineering. Da in jedem Erweiterungsschritt nur wenig neues Wissen hinzugefügt und danach die gesamte Wissensbasis erneut getestet wird, können Fehler und Widersprüchlichkeiten schnell erkannt und beseitigt werden.

Sofern die Shell bereits vorhanden ist, ist es möglich, in kurzer Zeit ein einfaches Prototyp-System zu erstellen und zu testen[3]. Auf diese Weise kann bereits frühzeitig geprüft werden, ob die Expertensystem-Technologie im Problembereich tatsächlich erfolgversprechend eingesetzt werden kann.

---

1) Vgl. Ernst, Ojha (20), S.182 f.
2) Waterman führt in (91), S.4 aus: "The problem-solving power of a program comes from the knowledge it posesses, not just from the formalisms and inference schemes it employs".
3) Diese Vorgehensweise wird auch als "Rapid Prototyping" bezeichnet. Vgl. z.B. Schmitz, Lenz (77), S.509.

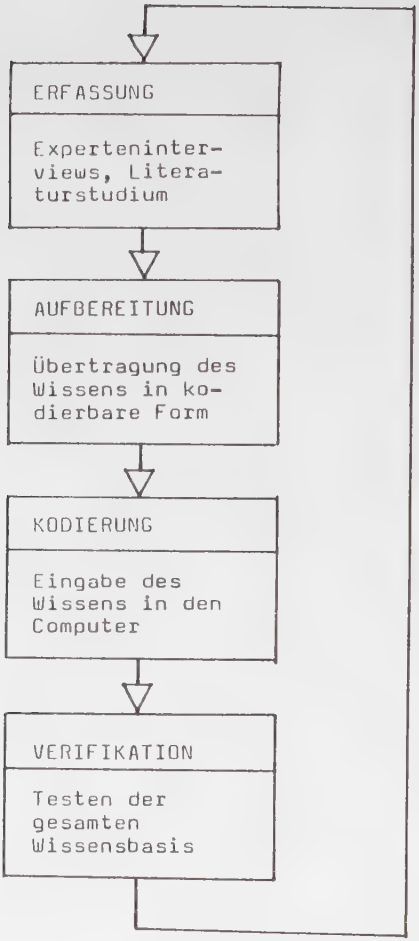


Abb. 3.2: Der Prozeß der Wissens-
akquisition

Zudem beinhaltet die inkrementelle Vorgehensweise einen erheblichen Motivationseffekt[1]. Der Fachgebiets Experte kann die Entwicklung des Systems beobachten und erfährt unmittelbar die Ergebnisse seiner Mitarbeit. Aus diesem Grund sollte der Experte erst dann wirklich aktiv in den Prozess der Wissensakquisition einbezogen werden, wenn die Shell fehlerfrei arbeitet und die grundlegenden Konzepte der Wissensdarstellung erarbeitet sind.

In der ersten Phase jedes Erweiterungsschrittes muß zunächst das hinzuzufügende Wissen erfaßt werden. Dies geschieht in der Regel durch Literaturstudium oder Expertenbefragung[2]. Das Wissen liegt nun in Form von natürlichsprachlichen verbalen Aussagen vor.

In der nächsten Phase wird das Wissen aufbereitet, d.h. in eine vom Expertensystem zu verarbeitende Form gebracht. Diese Form ist im zweiten Schritt der Erstellung des Expertensystems festgelegt worden und wird auch durch die Art der verwendeten Hilfsmittel bestimmt[3].

Danach wird das Wissen kodiert, d.h. in den Computer eingegeben. Dies geschieht mit Hilfe der Wissensakquisitionskomponente bzw. mit Hilfe eines Editors für die verwendete Programmiersprache.

---

1) Dieser Effekt konnte auch während der Erstellung des Expertensystems FOUNDER beobachtet werden: der anfangs aufgeschlossene, aber skeptische Experte entwickelte im Laufe der Zeit eine wachsende Begeisterung für das entstehende System. Vgl. auch Waterman (91), S.194.
2) Vgl. den folgenden Abschnitt.
3) Vgl. die Abschnitte 2.3 und 2.4.

Schließlich wird das Wissen verifiziert, d.h. es wird geprüft, wie sich das Verhalten des Systems durch Hinzufügen der neuen Elemente der Wissensbasis verändert. Dies geschieht in der Praxis meist durch Testfälle, d.h. durch reale Probleme, deren Lösung bereits bekannt ist[1]. Die Verifikation der Wissensbasis umfaßt die Prüfung auf Vollständigkeit, Richtigkeit und Widerspruchsfreiheit[2].

Vollständigkeit bedeutet dabei, daß das System eine den zuvor gesetzten Ansprüchen genügende Fehlerquote aufweist, denn meist ist es in der Praxis nicht möglich, eine sämtliches relevante Wissen umfassende Wissensbasis zu konstruieren. Richtigkeit bedeutet, daß das in der Wissensbasis enthaltene Wissen die Gegebenheiten des realen Problemfeldes sachlich richtig abbildet.

Widerspruchsfreiheit bedeutet, daß die neu hinzugefügten Wissenselemente dem bereits enthaltenen Wissen nicht widersprechen dürfen. Widersprüche können vor allem dann entstehen, wenn Wissen aus mehreren verschiedenen Quellen (z.B. verschiedene Experten oder Literaturmeinungen) verwendet wird. Im Forschungsbereich der künstlichen Intelligenz wird versucht, Methoden zu entwickeln, um solche Widersprüche automatisch aufzudekken. Dies ist bisher jedoch kaum in praktisch anwendbarer Weise gelungen[3].

Da selbstverständlich nicht alle denkbaren Problemfälle getestet werden können, ist es nicht möglich, die Vollständigkeit, Richtigkeit und Widerspruchsfreiheit der Wissensbasis zu garantieren. Aufgrund des sich in fast allen Wissensgebieten dynamisch verändernden Wissens

1) Zu den Methoden des Testens einer Wissensbasis vgl. Gaschnig u.a. (27), S.241 ff.
2) Vgl. Waterman (91), S.138.
3) Vgl. ebenda, S.92.

muß der Prozess des Erweiterns, Verbesserns und Testens auch während des praktischen Einsatzes fortgeführt werden, ähnlich wie sich auch ein menschlicher Experte ständig weiterbilden muß, um "auf dem neuesten Stand zu bleiben".

3.1.3 Methoden der Wissensakquisition

In der Praxis wird die Wissensakquisition in der Regel von einem sog. "Knowledge Engineer" oder "Wissensingenieur" vorgenommen[1]. Seine Aufgabe ist es, das unsystematisch und unstrukturiert vorliegende Expertenwissen für die Verarbeitung im Expertensystem aufzubereiten.

Die Anforderungen an den Knowledge Engineer umfassen tiefgehende Kenntnisse und Erfahrungen im Bereich der Expertensystem-Technologie und eine Einarbeitung in die grundlegenden Begriffe und Sachverhalte des jeweiligen Fachgebietes. Der Knowledge Engineer muß nicht selbst Experte für das jeweilige Fachgebiet sein; dies ist im Gegenteil eher hinderlich, da es dazu führen kann, daß die Schlußfolgerungsweise des Systems später, wenn überhaupt, nur von anderen Experten nachvollzogen werden kann[2]. Er muß aber mit dem Experten in dessen Fachsprache reden können und grundlegende Begriffe des Fachgebietes kennen, um die Expertenbefragung nicht durch langwierige Grundlagen-Erklärungen zu behindern.

---

1) Vgl. Noelke (62), S.109.
2) Waterman drückt dies durch den Ausspruch "Don´t be your own expert!" aus ((91), S.154).

Für den Erfolg der Wissensakquisition ist es sehr wichtig, daß der Experte über den gesamten Zeitraum des Projekts zur Mitarbeit motiviert ist[1]. Aus diesem Grund sollte der Knowledge Engineer den Experten in den Prozess der Erstellung des Expertensystems einbeziehen, indem er ihm die Grundlagen der Expertensystem-Technologie erläutert und ihn die Entstehung und Verbesserung des Systems beobachten läßt. Auf diese Weise wird auch die Grundlage für eine verbesserte Kommunikation zwischen Experte und Knowledge Engineer gelegt, da der Experte mit der Zeit lernt, worauf es bei der Formulierung von Wissen für ein Expertensystem ankommt.

Die beiden wichtigsten Quellen für die Wissensakquisition sind, wie schon angedeutet, die Fachliteratur und die Expertenbefragung. Dabei dient die Fachliteratur primär dem Zweck, die für eine erfolgreiche Expertenbefragung notwendigen begrifflichen Grundlagen zu erarbeiten. Weiterhin findet man dort oft Systematiken und Begriffshierarchien, die zur Strukturierung und Aufbereitung des Wissens hilfreich sind. Handbücher, Richtlinien, Handlungsanweisungen oder Entscheidungstabellen können je nach Fachgebiet ebenfalls sehr wertvolle Wissensquellen sein, da dort das Wissen oft schon in komprimierter und leicht formalisierbarer Form vorhanden ist.

Für die Expertenbefragung werden in der Literatur verschiedene Methoden vorgeschlagen[2]. Waterman[3] unterscheidet als wichtigste grundlegende Typen die "beobachtende" und die "intuitive" Vorgehensweise.

---

1) Vgl. z.B. Waterman (91), S.194.
2) Vgl. z.B. Noelke (62), S.121 f.; Buchanan u.a. (11), S.129 ff.
3) Vgl. Waterman (91), S.156 ff.

Das Wesen der beobachtenden Befragung liegt darin, daß dem Experten ein reales oder konstruiertes Problem aus dem Fachgebiet vorgelegt wird und seine Vorgehensweise zur Problemlösung "beobachtet" wird. Dazu muß der Experte bei der Lösung "laut denken", d.h. seine Gedankenschritte kurz erläutern. Erst nach Beendigung der Problemlösung geht der Knowledge Engineer mit gezielten Fragen auf die einzelnen Schritte näher ein und versucht, eventuelle Verständnislücken zu schließen.

Der wichtigste Vorteil der "beobachtenden" Befragung ist die Tatsache, daß ein Bild von der grundlegenden Vorgehensweise bei der Problemlösung erlangt wird. Der Nachteil liegt darin, daß die einzelnen Gedankenschritte oft intuitiv erfolgen und der Experte sie teilweise nicht auf Anhieb logisch begründen kann. Ein Nachhaken des Knowledge Engineers kann, wie empirische Studien zeigen, sogar dazu führen, daß der Experte falsche Begründungen für seine Vorgehensweise "erfindet"[1].

Bei der intuitiven Befragung wird demgegenüber der Experte mit Regeln oder Hypothesen über das Fachgebiet konfrontiert, die vom Knowledge Engineer oder einem anderen Experten aufgestellt wurden. Solche Aussagen haben beispielsweise die Form "Könnte man nicht sagen, daß aus Umstand X Faktum Y folgt...". Der Experte nimmt dann zu dieser Aussage Stellung und begründet ihre Richtigkeit oder Unrichtigkeit.

---

1) Diese Tatsache entspringt dem "Knowledge Engineering-Paradoxon": "Je kompetenter ein Experte, desto unfähiger ist er, seine Schlußfolgerungsweise zu beschreiben". Vgl. Waterman (91), S.154.

Die intuitive Methode hat den Vorteil, daß der Experte Zusammenhänge und Regeln des Fachgebietes oft besser darstellen kann, wenn er mit entsprechenden Aussagen konfrontiert wird. Der Nachteil der intuitiven Methode liegt darin, daß der Gesamtzusammenhang der Problemlösung nur unzureichend erfaßt ist. Weiterhin besteht die Gefahr, daß wichtige Aspekte der Problemlösung übersehen werden.

Die Vor- und Nachteile der Methoden zeigen, daß es sinnvoll ist, beide Vorgehensweisen in einem Interview zu kombinieren. Dazu könnte dem Experten z.B. zunächst eine Fallstudie vorgelegt und die beobachtende Methode angewandt werden. Anschließend könnte der Knowledge Engineer dem Experten Konzepte und Hypothesen vorlegen, die er aus der Beobachtung erkannt zu haben glaubt. Gut geeignet für eine Expertenbefragung ist auch die Analyse einer falschen Lösung des Prototyp-Expertensystems. Hier kann der Experte durch Untersuchen der fehlerhaften Schlußfolgerungsweise Lücken in der Wissensbasis erkennen und schließen.

Tabelle 3.1 fasst die wichtigsten Befragungsmethoden mit einer Kurzbeschreibung zusammen[1]. Es handelt sich dabei um verschiedene Ausprägungen der beiden beschriebenen grundlegenden Vorgehensweisen. In der Spalte "Typ" ist die Tendenz der Methode zu eher intuitiver oder beobachtender Vorgehensweise bzw. einer Mischung von beidem angegeben.

---

1) Vgl. Waterman (91), S.156 ff., der dort die hier nur stichwortartig beschriebenen Methoden näher erläutert.

Methode	Typ	Kurzbeschreibung
"On-the-Job"-Beobachtung	beobachtend	K.E. beobachtet Experten im täglichen Einsatz
Problem-Diskussion	intuitiv	Informelle Diskussion über ein praktisches Problem
Problem-Beschreibung	beobachtend	Experte beschreibt für eine gegebene Lösung typische Problemstellungen
Problem-Analyse	beobachtend	Experte beschreibt Vorgehensweise während der Lösung eines Problems ("Lautes Denken")
System-Verbesserung	intuitiv	Experte stellt Fallstudie, die von K.E. bzw. Expertensystem mit vorhandenem Wissen gelöst wird
System-Beobachtung	intuitiv	Experte beobachtet und kritisiert Expertensystem bei der Arbeit
System-Bestätigung	beobachtend/ intuitiv	System und Wissensbasis werden bisher unbeteiligten Experten zur Kontrolle vorgelegt

Tabelle 3.1 : Methoden der Expertenbefragung nach Waterman

3.1.4 Probleme der Wissensakquisition

In der Praxis treten bei der Wissensakquisition verschiedene Probleme auf, die die Konstruktion der Wissensbasis empfindlich stören können. Hier soll vor allem auf solche Probleme eingegangen werden, die aus der Interaktion zwischen Knowledge Engineer und Experte entstehen. Probleme, die aufgrund unzureichender Kenntnisse des Knowledge Engineers oder unadäquater Hilfsmittel entstehen, werden nicht näher untersucht.

Probleme der Interaktion von Experte und Knowledge Engineer können auf verschiedenen Ebenen auftreten. Einer der wichtigsten Problembereiche ist die Bereitschaft zur Mitwirkung von der Seite des Experten[1]. Es ist unbedingt notwendig, daß diese Bereitschaft vorhanden ist, da niemand zur Preisgabe seines Wissens gezwungen werden kann und eine sinnvolle Zusammenarbeit unter Druck nicht denkbar ist. Mangelnde Motivation des Experten kann ihre Ursachen in Angst, Skepsis oder persönlichen Interessen haben.

Angst kann sich darauf gründen, daß befürchtet wird, das Expertensystem könnte den Experten ersetzen oder sein möglicherweise mangelndes Fachwissen könnte während des Erstellungsprozesses aufgedeckt werden. Hier ist besonders wichtig, daß der Knowledge Engineer sehr behutsam vorgeht. Dem Experten muß klargemacht werden, daß das Expertensystem ihn und andere bei der täglichen Arbeit unterstützen soll und daß Expertensysteme menschliche Experten aus in Abschnitt 3.2 noch zu erläuternden Gründen nicht ersetzen können und sollen[2].

1) Vgl. Waterman (91), S.194.
2) Vgl. Bell (3), S.618.

Gegebenenfalls muß klargestellt werden, daß es bei einer Expertenbefragung nicht auf "Schulbuch" bzw. "Prüfungswissen", sondern auf praktische Erfahrung und "Know How" ankommt. Aus diesem Grunde sollte ein Experte, der fähig ist, praktische Probleme besser als Laien zu lösen, niemals befürchten müssen, schlecht beurteilt zu werden, weil er nicht auf dem neuesten theoretischen Stand des Fachgebietes ist.

Skepsis gegenüber Expertensystemen kann sich z.B. auf schlechte Erfahrungen mit anderen Computer-Anwendungen gründen[1]. Sie tritt vor allem dann auf, wenn überzogene Erwartungen an die Leistungsfähigkeit der Expertensysteme gestellt und öffentlich erklärt werden[2]. Dem Experten muß eine realistische Zielsetzung für das Expertensystem vorgelegt werden. Sofern vorhanden, können Beispiele von Expertensystemen, die in verwandten Gebieten erfolgreich eingesetzt wurden, das Mißtrauen abbauen. Eine "gesunde" Skepsis vor allem zu Beginn ist jedoch natürlich und eher vorteilhaft. Sie baut sich in der Regel im Verlauf des Projektes von selbst ab.

Teilweise stehen auch persönliche Motive einer aktiven Mitarbeit bei der Konstruktion des Expertensystems entgegen[3]. Dazu zählt vor allem die Angst, eine "Monopolstellung" für das Expertenwissen einzubüßen und innerhalb des Arbeitsumfeldes an Bedeutung zu verlieren[4]. Dem kann entgegengewirkt werden, indem deutlich gemacht wird, daß der Experte für die Aufgabe seiner alten Monopolstellung eine neue erhält, indem er

---

1) Vgl. Waterman (91), S.194 f.
2) Vgl. Abschnitt 2.1.1.
3) Vgl. Bell (3), S.613.
4) von Zelewski spricht in diesem Zusammenhang von "kognitiver Enteignung" ((99), S.1161).

"Experte für Expertensysteme" wird. Gegebenenfalls kann die Motivation auch durch finanzielle Anreize oder Aufstiegschancen erhöht werden.

Ein weiterer wichtiger Faktor, der die Konstruktion der Wissensbasis behindern kann, ist die mangelnde zeitliche Freistellung des Experten[1] für die Befragungen. Dies kann, z.B. aufgrund mangelnder Motivation, durch den Experten selbst oder durch seine Vorgesetzten geschehen.

Sofern der Experte zur Mitarbeit genügend motiviert ist, können ihn gravierende Probleme oder ein hohes Pensum in seinem Arbeitsbereich dennoch von der Mitarbeit abhalten. Der Knowledge Engineer muß darauf achten, daß die Befragungen nicht in Spitzen-Belastungszeiten (z.B. aufgrund saisonaler Schwankungen) liegen. Auf keinen Fall darf die Befragung zu einer zusätzlichen Belastung für den Experten werden, da dies seine Motivation rasch zerstören würde.

Von Seiten des Managements muß dafür gesorgt werden, daß der Experte für einen ausreichenden Zeitraum von Routinetätigkeiten entlastet wird. Dem steht in der Regel die überlegung entgegen, daß jede Stunde, in der ein Experte nicht für seine normalen Aufgaben zur Verfügung steht, hohe Kosten verursacht. Man sollte dies von vornherein in die Entscheidung über den Einsatz der Expertensystem Technologie einkalkulieren und dabei berücksichtigen, daß sich bei Erfolg des Projektes später in der Regel ein erheblicher Produktivitätszuwachs ergibt. Ein gescheiterter Versuch ist in jedem Fall teurer als einige zusätzliche Befragungsstunden.

1) Vgl. Bell (3), S.613.

Ebenfalls wichtig ist, daß dem Experten durch seine Mitarbeit bei der Wissensakquisition keine Nachteile entstehen. Solche Nachteile könnten ihre Ursachen z.B. in einem betriebswirtschaftlichen Ergebnisvergleich zwischen verschiedenen Kostenstellen haben, sofern der Experte für das Ergebnis einer dieser Kostenstellen die Verantwortung trägt. Er könnte befürchten, daß durch seinen Einsatz für die Wissensakquisition und die damit verbundene zeitliche Beschränkung das Ergebnis seines Verantwortungsbereichs verschlechtert wird. Eine derartige Befürchtung würde die Motivation zur Mitarbeit erheblich beeinträchtigen.

Der dritte Störfaktor für die Wissensakquisition sind Schwierigkeiten in der Kommunikation zwischen Experte und Knowledge Engineer. Sie können ihre Ursachen in mangelnden Vorkenntnissen des Knowledge Engineers oder unzureichender Fähigkeit des Experten, Zusammenhänge verständlich zu erklären[1], haben. Der Knowledge Engineer kann diesen Schwierigkeiten durch hohe Einfühlsamkeit und gründliche Vorbereitung begegnen.

Werden später erhebliche Wissenslücken auf der Seite des Knowledge Engineers festgestellt, sollte eine Pause in den Befragungen eingelegt werden, in der er die Lükken, z.B. durch Literaturstudium, ausfüllen kann. Dadurch lassen sich fruchtlose und demotivierende Befragungssitzungen vermeiden. Sofern eine Auswahl besteht, sollte bereits bei der Wahl des Experten auf dessen Bereitschaft und Fähigkeit zur Kommunikation geachtet werden[2].

1) Vgl. Waterman (91), S.192.
2) Vgl. ebenda.

Weitere Probleme können aus der Befragung mehrerer verschiedener Experten resultieren''. Dieses Vorgehen hat den Vorteil, subjektive Einschätzungen und Wissenslücken eines Experten zum großen Teil auszugleichen. Allerdings birgt es die Gefahr von Widersprüchen in der Wissensbasis. Weitere Schwierigkeiten resultieren aus den Beziehungen der Experten untereinander, die u.a. von Konkurrenzdenken geprägt sein können. Insgesamt erscheint es vorteilhaft, nur einen verantwortlichen Experten als Hauptträger der Wissensakquisition zu benennen und weitere Experten vor allem zur Verifikation des Systems hinzuzuziehen.

1) Vgl. Waterman (91), S.195.

3.2 Spezialprobleme der Expertensystem-Technologie

3.2.1 Das Akzeptanzproblem

Ein besonderes Problem der Anwendung der Expertensystem-Technologie ergibt sich aus der geforderten Akzeptanz durch den Benutzer[1]. Vertraut der Benutzer dem Expertensystem nicht und wendet es daher auf praktische Probleme nicht an, so war der gesamte Entwicklungsaufwand vergeblich.

Aus der Forderung nach Akzeptanz durch den Benutzer ergeben sich neben einer gründlichen Einweisung des Benutzers in Möglichkeiten und Grenzen des Systems folgende Anforderungen an das Expertensystem:

- Das Expertensystem muß im praktischen Gebrauch zuverlässig sein, d.h. die Fehlerrate muß auf ein vertretbares und unvermeidbares Maß reduziert werden[2]. Fehler sollten nur bei ungewöhnlichen Problemstellungen auftreten; alltägliche Routineprobleme sollte das System in jedem Fall fehlerfrei lösen können. Die Zuverlässigkeit kann nur durch eine sorgfältig konstruierte und laufend gepflegte Wissensbasis gewährleistet werden.

---

1) Vgl. Bell (3), S.617 f. Von Zelewski analysiert in (99), S.1282 ff. das Akzeptanzproblem ausführlich hinsichtlich seiner sozialen Komponente, d.h. der Angst der Benutzer vor negativen Auswirkungen des Expertensystems auf ihr unmittelbares Arbeitsumfeld. Demgegenüber wird hier vor allem auf den Aspekt der Nützlichkeit des Systems für den Benutzer Bezug genommen.
2) Vgl. Bell (3), S.618.

Die Schlußfolgerungen des Systems müssen nachvollziehbar sein. Dies ist erforderlich, um eine Kontrolle der Problemlösung durch einen Experten besonders bei ungewöhnlichen Problemstellungen zu ermöglichen bzw. zu erleichtern. Weiterhin sollten eventuelle Fehler leicht lokalisierbar und korrigierbar sein. Aus diesen Forderungen ergibt sich die Notwendigkeit einer komfortablen und aussagekräftigen Erklärungskomponente[1].

Das System muß eine den praktischen Anforderungen entsprechende Benutzeroberfläche aufweisen[2]. Dies erfordert nicht nur eine möglichst einfache Bedienbarkeit der einzelnen Komponenten der Shell, sondern auch kurze Antwortzeiten des Systems. Zu lange Rechenzeiten zwischen den Eingaben behindern die Konzentration des Benutzers und beeinträchtigen die Akzeptanz des Systems erheblich.

Fragen des Systems an den Benutzer müssen eindeutig und verständlich formuliert sein. Zudem sollte der Grund für eine Frage jederzeit erfragbar sein, damit sie dem Zusammenhang entsprechend beantwortet werden kann. Ein wesentliches Hindernis für die Akzeptanz sind zu viele bzw. zu wenig zielgerichtete Fragen. Insbesondere bei großen Wissensbasen sollte das System über eine zielstrebige und geschickte Schlußfolgerungsstrategie verfügen und nur den jeweils aussichtsreichsten Lösungsweg weiterverfolgen.

---

1) Vgl. Roesner (72), S.44.
2) Vgl. ebenda, S.41 f.

Zusammenfassend läßt sich sagen, daß eine hohe Akzeptanz durch den Benutzer nur dann zu erreichen ist, wenn das System in der Praxis zu einer wesentlichen Arbeitserleichterung führt. Dies betrifft besonders die Zuverlässigkeit: die Arbeits- und Zeitersparnis durch die Anwendung des Systems darf nicht durch den Aufwand zur Behebung oder Verhinderung von Fehlern bzw. durch deren Folgekosten zunichte gemacht werden. Dies unterstreicht noch einmal die Bedeutung einer sorgfältigen Konstruktion und Pflege der Wissensbasis.

3.2.2 Das Verantwortlichkeits-Problem

Im Zusammenhang mit der praktischen Anwendung von Expertensystemen taucht immer wieder ein Problem auf, das die Anwendungsmöglichkeiten teilweise erheblich einschränkt. Es ist das Problem der Verantwortung für Entscheidungen, insbesondere Fehlentscheidungen aufgrund von Empfehlungen oder Einschätzungen des Expertensystems[1].

Aus dieser Fragestellung ergibt sich zunächst ein zivilrechtliches Problem: Wer haftet für Schäden, die aus fehlerhaften Antworten des Expertensystems entstehen? Obwohl diese Frage juristisch noch nicht völlig geklärt[2] und auch nicht Gegenstand dieser Arbeit ist, ist abzusehen, daß die Hersteller bzw. Konstrukteure von Expertensystemen, vor allem aufgrund der bereits angeführten Unmöglichkeit einer vollständigen Wissensbasis, keine Garantie für Fehlerfreiheit geben und somit auch keine Haftung übernehmen können. Das Fehlerrisiko bleibt also tendenziell beim Anwender.

---

1) Vgl. Bell (3), S.618, der dort erwähnt, daß wenigstens ein Entwicklungsprojekt im Bereich der Medizin aufgrund dieser Problematik abgebrochen worden sei.
2) Vgl. Zeide, Liebowitz (98), S.19 ff.

Unabhängig von der juristischen Problematik stellt sich innerhalb betrieblicher Organisationen ein ähnliches Problem. Einerseits muß im Interesse einer Dezentralisierung von Entscheidungskompetenzen die Verantwortung für einzelne Entscheidungsbereiche an untergeordnete Instanzen übertragen werden[1]. Andererseits können diese Instanzen jedoch nicht in vollem Umfang für Fehler aufgrund von falschen Antworten eines Expertensystems verantwortlich gemacht werden, wenn sie diese nicht erkennen und somit Fehlentscheidungen nicht verhindern konnten.

Aus diesem Grund ergibt sich die Notwendigkeit, daß der Benutzer die Vorschläge des Expertensystems überprüfen bzw. ihre Richtigkeit beurteilen kann. Dies ist jedoch nur möglich, wenn der Benutzer das System und seine Schlußfolgerungen versteht. Neben den im vorherigen Abschnitt erhobenen Forderungen ergibt sich daraus, daß der Benutzer über ein gewisses Maß an Fachwissen verfügen muß.

Aus dieser Erkenntnis folgt eine gravierende Einschränkung der Anwendungsmöglichkeiten von Expertensystemen. Expertensysteme können als Computerprogramme keine Entscheidungsverantwortung übernehmen. Sie können daher auch menschliche Entscheidungskompetenz nicht ersetzen. Stattdessen muß ihr Einsatz darauf beschränkt bleiben, den menschlichen Entscheidungsträger bei seiner Entscheidungsfindung zu <u>unterstützen</u>[2].

---

1) Vgl. z.B. Schierenbeck (76), S.99.
2) Vgl. Luconi, Malone, Morton (45), S.6 ff.

Es ist also nicht sinnvoll, an die Stelle eines menschlichen Experten ein Expertensystem mit einem Laien als Benutzer treten zu lassen. Stattdessen kann jedoch die Entscheidungskompetenz eines Verantwortungsträgers wesentlich erhöht werden, indem ihm selten benötigtes oder schwer zugängliches und umfangreiches Expertenwissen bzw. die Erfahrungen von Spezialisten zugänglich gemacht werden.

3.2.3 Die Bedeutung von Allgemeinwissen für die Problemlösung

Ein weiterer Grund dafür, daß Expertensysteme Menschen bei der Problemlösung nicht ersetzen können, ist ihre Unfähigkeit, Allgemeinwissen zu verarbeiten[1]. Unter Allgemeinwissen soll dabei solches Wissen verstanden werden, das nicht zu dem jeweiligen Fachgebiet gehört, für die Problemlösung aber dennoch relevant sein kann. Dazu gehören Fakten über die allgemeine Umwelt wie auch die Fähigkeit, die Grenzen des eigenen Wissens zu beurteilen.

An einem Beispiel[2] läßt sich demonstrieren, wie Allgemeinwissen benutzt werden kann, um Fehler in Daten zu erkennen und zu korrigieren:

In einem Expertensystem zur medizinischen Diagnose mögen die Daten für einen Patienten folgendermaßen angegeben worden sein: Gewicht: 14 Pfund, Alter: 110 Jahre. Ein Mensch wird beim Lesen dieser Daten sofort stutzig: zwar können Menschen sowohl 110 Jahre alt werden als auch 14 Pfund wiegen - die Kombination beider Werte ist

1) Vgl. von Zelewski (99), S.893.
2) Vgl. Waterman (91), S.15.

jedoch praktisch ausgeschlossen. Der Mensch würde vermuten, daß beide Werte bei der Eingabe vertauscht worden sind und es sich in Wirklichkeit um ein 14 Jahre altes Kind mit einem Gewicht von 110 Pfund handelt.

Um die oben genannte Schlußfolgerung zu ziehen, die keinerlei medizinische Vorbildung verlangt, müßte ein Expertensystem über eine Tabelle aller möglichen Beziehungen zwischen Gewicht und Alter sowie über Wissen darüber verfügen, daß das Vertauschen zweier Daten bei der Eingabe eine häufige Fehlerquelle ist. Dies ist in der Praxis aufgrund der zur Berücksichtigung aller derartigen Fehler erforderlichen Datenmengen praktisch ausgeschlossen.

Ein weiteres Beispiel[1] verdeutlicht die Bedeutung der Fähigkeit, das eigene Wissen einzuschätzen: Fragt man einen Europäer nach der Telefonnummer des Staatspräsidenten von Uganda, so wird er in der Regel nicht sein Gedächtnis nach der Nummer durchsuchen. Er weiß vielmehr, daß er die Telefonnummer nicht kennt. Dieses Wissen über das eigene Unwissen ist in einem Expertensystem mit der heutigen Technologie praktisch nicht abzubilden.

Ebenso bedeutungsvoll ist die Beurteilung der Lösungsmöglichkeit eines Problems. Niemand würde beispielsweise auf die Idee kommen, ein Telefonbuch oder die Fernsprechauskunft nach der Telefonnummer von Wolfgang Amadeus Mozart zu befragen. Wir wissen, daß es hierfür keine Lösung gibt, weil wir wissen, daß es zu Mozarts Lebzeiten keine Telefone gab. Ein Expertensystem würde unter Umstanden sämtliche verfügbaren Datenbanken nach der relevanten Information durchsuchen.

1) Vgl. Waterman (91), S.15.

Die Unfähigkeit von Expertensystemen, ihre eigene Lösungsfähigkeit zu beurteilen, kann - im günstigeren Fall - dazu führen, daß durch fortgesetztes Befragen des Benutzers ein Problem zu lösen versucht wird, bei dem ein Experte mit einem Blick erkannt hätte, daß keine Lösung existiert. Im ungünstigeren Fall wird eine falsche Lösung erzeugt, dem Benutzer gegenüber jedoch der Eindruck erweckt, das Expertensystem habe alle relevanten Gesichtspunkte berücksichtigt[1]. Mit M.Z.Bell könnte man sagen: "Wenig Wissen ist manchmal schlimmer als gar keines"[2].

Im Bereich betriebswirtschaftlicher Probleme können z.B. gesamtwirtschaftliche oder gesellschaftliche Rahmenbedingungen einen wesentlichen Einfluß auf Problemlösungen haben. Wenn das Expertensystem diese relevanten Größen nicht erfassen oder beurteilen kann, wird es zwangsläufig zu falschen Schlußfolgerungen kommen. Dennoch können diese Schlußfolgerungen einen wesentlichen Erkenntniswert für den Entscheidungsträger beinhalten, wenn dieser nämlich in der Lage ist, sie vor dem Hintergrund der jeweiligen Situation richtig zu interpretieren[3]. In diesem Fall wird das Allgemeinwissen des Entscheidungsträgers mit dem Spezialwissen des Expertensystems zu einer leistungsfähigen Einheit verknüpft.

Weitere Eigenschaften menschlicher Experten, die von Expertensystemen (noch) nicht nachempfunden werden können, sind die Kreativität[4], d.h. die Fähigkeit, z.B. durch Analogien, neue Lösungswege zu erfinden, und die Fähigkeit, aus Fehlern zu lernen[5]. Während er

---

1) Vgl. das Beispiel in Bell (3), S.614 ff.
2) Vgl. ebenda, S.618.
3) Vgl. ebenda.
4) Vgl. Waterman (91), S.14.
5) Vgl. ebenda.

stere wohl auf absehbare Zeit in Computern nicht abgebildet werden kann, wird auf dem Gebiet des automatischen Lernens intensiv geforscht[1]. Bisher sind jedoch kaum für Expertensysteme praktisch verwendbare Ergebnisse erzielt worden.

Zusammenfassend kann gesagt werden, daß sich aus der Unfähigkeit von Expertensystemen, Allgemeinwissen zu verarbeiten, die Notwendigkeit einer Verknüpfung des Spezialwissens des Expertensystems mit der menschlichen Fähigkeit, Gesamtzusammenhänge zu erkennen und Werte vor diesem Hintergrund richtig zu interpretieren, ergibt. Hieraus folgt, daß ein Expertensystem nicht unabhängig von menschlichem Allgemeinwissen eingesetzt werden kann.

1) Vgl. zum Stand der Forschung auf diesem Gebiet z.B.Pohmann (21).

3.3 Anwendungsbereiche für Expertensysteme

3.3.1 Anwendungsvoraussetzungen

Die Voraussetzungen zur Anwendung der Expertensystem Technologie auf ein gegebenes Problem umfassen eine instrumentelle, eine personelle und eine strukturelle Komponente. Alle drei Komponenten müssen erfüllt sein, damit die erfolgreiche Erstellung eines Expertensystems möglich wird.

Die instrumentelle Komponente beinhaltet die Hard- und Software, mit deren Hilfe das Expertensystem erstellt und betrieben werden soll. Hinsichtlich der Hardware ergibt sich die Forderung nach einer ausreichenden Rechengeschwindigkeit[1] sowie einem ausreichenden Speicherplatz, damit auch eine im Laufe der Zeit vergrößerte Wissensbasis noch verwaltet werden kann.

Die Software sollte die Erstellung des Expertensystems möglichst umfassend unterstützen. Sie kann je nach Anwendung aus einer Shell[2], einer Knowledge Engineering Sprache[3] oder einer allgemeinen Programmiersprache[4] sowie Hilfsprogrammen zur Dateiverwaltung oder Fehlersuche bestehen. Die Software muß so beschaffen sein, daß das Wissen in der für die jeweilige Problemstellung erforderlichen Weise dargestellt werden kann. Aus diesem Grunde ist zur Auswahl der Hard- und Software in der Regel bereits eine Kenntnis der grundlegenden Problemstruktur notwendig.

---

1) Vgl. Abschnitt 3.2.1 sowie die Ausführungen zu "Echtzeitproblemen" im nächsten Abschnitt.
2) Vgl. Abschnitt 2.4.1.
3) Vgl. Abschnitt 2.4.2.
4) Vgl. Abschnitt 2.4.3.

Das fertige Expertensystem muß schließlich in die spätere Anwendungsumgebung integriert werden. Diese Forderung ist in gewissem Maße konfliktär zu der Forderung nach weitgehender Unterstützung durch die Software, da die meiste für die Erstellung von Expertensystemen entwickelte Software nur auf speziell dafür geeigneten Rechnern, sog. "LISP-Maschinen", angewandt werden kann. Diese Hardware-Software-Systeme können den Aufwand zur Erstellung eines Expertensystems wesentlich reduzieren. Es ist jedoch in aller Regel im nachhinein notwendig, das erstellte System auf die im Betrieb vorhandene Hardware zu übertragen. Aus diesem Grund wird vielfach versucht, das Expertensystem direkt auf der vorhandenen Hardware zu entwickeln und dabei ein Nachteil in der Unterstützung durch das System in Kauf genommen.

Für die Entwicklung eines Prototyp-Systems, das die Frage beantworten kann, ob eine weitere Expertensystem-Entwicklung im betrachteten Bereich lohnend ist, sind wesentlich geringere Anforderungen an Hard- und Software zu stellen. Hier ist bereits vielfach ein Personal-Computer ausreichend, für den einfachere, vorgefertigte Shells erhältlich sind. In diese Klasse gehört auch das in Abschnitt 2.5 vorgestellte System MOSES. Auf diese Weise ist es möglich, ohne größere Investitionen zunächst Erfahrungen im Umgang mit Expertensystemen zu sammeln.

Die personelle Komponente der Voraussetzungen umfaßt die Forderung nach der Verfügbarkeit wenigstens eines Experten sowie eines mit der Expertensystem-Technologie vertrauten Knowledge Engineers. Vor allem die erste Forderung ist keineswegs trivial. Sie umfaßt die Voraussetzung, daß ein Experte existieren muß, der das betrachtete Problem wesentlich besser als ein Laie lösen kann, daß dieser zeitlich in ausreichendem Maße zur Verfügung steht und daß er für die Mitarbeit an dem

System ausreichend motiviert ist[1]. Der Experte muß zudem in der Lage sein, das bei der Problemlösung verwandte Wissen zu artikulieren.

Im Bereich betriebswirtschaftlicher Fragestellungen kann der Experte in der Regel nur aus dem Unternehmen selbst kommen, da nur in diesem Fall gewährleistet ist, daß wirklich alle im betrachteten Unternehmen relevanten Zusammenhänge erkannt und berücksichtigt werden. Steht unternehmensintern kein erfahrener Knowledge Engineer zur Verfügung, kann auf ein externes Beratungsunternehmen zurückgegriffen werden.

Die dritte Komponente der Voraussetzungen betrifft die Struktur des zugrundeliegenden Problems. In der Betriebswirtschaftslehre werden wohlstrukturierte und schlechtstrukturierte Probleme unterschieden[2]. Bei wohlstrukturierten Problemen ist erstens ein scharf definierter Lösungsraum vorhanden; zweitens existiert eine operationale Zielfunktion, die es erlaubt, die beste Lösung aus dem Lösungsbereich eindeutig zu bestimmen; und drittens existiert ein Lösungsverfahren, mit dem die beste Lösung in vertretbarer Zeit ausgewählt werden kann. Wohlstrukturierte Probleme kommen für die Entwicklung von Expertensystemen nicht in Frage, weil diese zu aufwendig wäre und eine einfachere Lösungsmethode ohnehin bekannt ist.

Demgegenüber erfüllen schlechtstrukturierte Probleme eine oder mehrere der oben genannten Bedingungen nicht. Ist nur die dritte Bedingung nicht erfüllt, d.h. es existiert keine effiziente Lösungsmethode, so spricht man von lösungsdefekten[3] Problemen. In diesem Fall

1) Vgl. Abschnitt 3.1.4.
2) Vgl. z.B. Newell, Simon (60); Simon (81), S.181 ff.; Adam (1), S.13 ff.; Witte (95), S.72 ff.
3) Vgl. Witte (95), S.76 ff.

werden üblicherweise Heuristiken angewandt, mit denen wenigstens eine verhältnismäßig gute, nicht notwendig jedoch die optimale Lösung aus dem Lösungsraum ausgewählt wird. Da Expertensysteme als komplexe Systeme von Heuristiken angesehen werden können, ist eine Anwendung auf lösungsdefekte Probleme prinzipiell möglich. Voraussetzung ist allerdings, wie schon angeführt, daß ein Experte existiert, der die entsprechenden Heuristiken und ihre Wirkungen kennt.

Wenn die zweite Bedingung nicht erfüllt ist, d.h. keine operationale Zielfunktion vorliegt, handelt es sich um ein zielsetzungsdefektes[1] Problem. In diesem Fall kann keine eindeutige Ordnung der Elemente des Lösungsraumes vorgenommen werden, z.B. weil widersprüchliche Zielsetzungen vorliegen. Ein Beispiel für ein solches Problem ist die Wertpapier-Anlageberatung. Hier konkurrieren die Zielsetzungen "hohe Rendite", "schnelle Liquidierbarkeit" und "geringes Risiko" miteinander. Eine begründete Auswahl kann nur getroffen werden, wenn es gelingt, die unterschiedlichen Bewertungskriterien zu gewichten oder in eine Rangfolge zu bringen und so den Zielsetzungsdefekt zu überwinden.

Expertensysteme können Zielsetzungskonflikte nicht lösen. Sie können jedoch benutzt werden, um alle möglichen Lösungen zu finden, und die Auswirkungen jeder Lösung auf die konkurrierenden Ziele aufzeigen. Auf diese Weise wird dem Benutzer eine Entscheidung auf der Basis "transparenter" Informationen ermöglicht. Im Gegensatz dazu wird bei den meisten analytischen Methoden eine

---

1) Vgl. Adam (1), S.16 f. Demgegenüber führt Witte in (95), S.76 ff., zielsetzungsdefekte Probleme nicht als eigene Problemklasse auf, sondern zählt sie zu den bewertungsdefekten Problemen.

eindeutige Zielfunktion gefordert, bei der im nachhinein nicht mehr festgestellt werden kann, wie ein bestimmter Zielfunktionswert zustandegekommen ist.

Expertensysteme können auch dazu eingesetzt werden, dem Benutzer beim Gewichten der Bewertungskriterien zu helfen. Im Beispiel der Wertpapier Anlageberatung könnte das System z.B. erfragen, ob die Anlage zum Zweck der Rücklagenbildung oder eher aus spekulativen Motiven heraus erfolgen soll. Im ersten Fall würde z.B. das Kriterium "geringes Risiko" hoch, im zweiten Fall niedrig gewichtet werden.

Ein bewertungsdefektes[1] Problem ist gegeben, wenn unklar ist, welches Element des Lösungsraumes der gegebenen Zielsetzung am besten entspricht, z.B. weil sich die Auswirkungen einer Entscheidung hinsichtlich der Zielgröße nicht quantifizieren lassen. Im Fall der Wertpapier-Anlageberatung könnte z.B. unklar sein, wie hoch das Risiko eines bestimmten Wertpapieres zu bewerten ist.

Der Bewertungsdefekt kann von einem Expertensystem nicht aufgehoben werden, da Wissen über die Vorziehenswürdigkeit einzelner Alternativen in einer spezifischen Problemsituation in der Wissensbasis zumindest implizit enthalten sein muß. Dennoch können Expertensysteme bei der Lösung bewertungsdefekter Probleme eingesetzt werden, indem die möglichen Lösungen aufgezeigt und der Bewertungsdefekt explizit dargestellt werden. Die Lösung des Bewertungsproblemes bleibt dann dem Benutzer überlassen.

1) Vgl. Witte (95), S.78 ff.

Ein Vorteil der Expertensystem-Technologie ist es, daß eine Bewertung nicht notwendig quantitativ erfolgen muß. Während analytische Modelle in der Regel genaue Zahlenwerte für die zu bewertenden Größen benötigen, können Expertensysteme auch qualitative Bewertungen wie "gut" oder "ausreichend" verarbeiten. So könnte eine Regel beispielsweise lauten: "Wenn die Risikobereitschaft des Anlegers hoch ist, dann ist eine Aktie empfehlenswert". Die Regel wäre in diesem Fall anwendbar, wenn der Benutzer seine Risikobereitschaft als "hoch" (und nicht z.B. als "sehr hoch" oder "überdurchschnittlich") einstuft[1]. Damit würde allerdings nicht das Bewertungsproblem gelöst, sondern lediglich das Quantifizierungsproblem umgangen.

Ein wirkungsdefektes[2] Problem liegt schließlich vor, wenn die Zusammenhänge zwischen Entscheidungen und ihren Folgen nicht bekannt sind. Dies ist in der Regel dann der Fall, wenn der Problembereich nicht hinreichend erforscht ist. Ob in diesem Fall ein Expertensystem erstellt werden kann, hängt davon ab, ob die Konsequenzen von Entscheidungen bzw. die Beziehungen zwischen beobachteten Tatbeständen und ihren Ursachen völlig unbekannt oder nur unsicher sind.

Im ersten Fall liegen überhaupt keine Informationen darüber vor, welche Konsequenz eine bestimmte Entscheidung bzw. welche Ursache ein Tatbestand hat, d.h. der Lösungsraum des Problems ist unbekannt. In diesem Fall

---

1) Qualitative Bewertungen können z.B.auch mit Hilfe der in Abschnitt 2.3.5 vorgestellten Theorie der unscharfen Mengen verarbeitet werden. Dadurch wäre eine feinere Abstufung der Bewertung als im Beispiel möglich. Aller- dings handelt man sich dadurch erneut das Problem ein, Zugehörigkeitsfunktionen quantitativ zu bestimmen.
2) Vgl. Witte (95), S.80 ff.

können auch keine Experten existieren, die das Problem lösen können, d.h. eine Anwendung der Expertensystem Technologie ist unmöglich.

Im zweiten Fall werden zwar die Zusammenhänge zwischen Entscheidung und Konsequenz bzw. zwischen Tatbestand und Ursache nicht genau verstanden, es liegen jedoch Erfahrungswerte darüber vor, daß bei einer bestimmten Entscheidung bzw. einem Tatbestand häufig eine bestimmte Wirkung bzw. Ursache zu erwarten ist. In diesem Fall ist der Lösungsraum nicht scharf begrenzt. Es existieren zwar mögliche Lösungen, es ist aber nicht sicher, ob sie in einem bestimmten Anwendungsfall zum Lösungsraum gehören oder nicht.

Da die kausalen Zusammenhänge nicht bekannt sind, können derartige Probleme nur mit Heuristiken, die auf Erfahrungen bzw. Beobachtungen beruhen, gelöst werden. In diesem Fall existieren in der Regel Experten, die das Problem anhand derartiger Heuristiken lösen können. Ein Expertensystem ist daher prinzipiell erstellbar.

Ein Beispiel für ein teilweise wirkungsdefektes Problem ist die Bilanzanalyse. Dabei kann aus der Entwicklung bestimmter Bilanzpositionen auf ein betriebliches Problem geschlossen werden, für das diese Entwicklung symptomatisch ist. Es ist jedoch nicht sicher, ob die vermutete Ursache tatsächlich gegeben ist oder ob die beobachtete Entwicklung auf andere, z.B. unternehmensspezifische Gründe zurückzuführen ist, da Informationen hierüber dem Bilanzanalytiker nicht zugänglich sind.

Neben den angeführten Defekten spielt eine weitere Determinante der Problemstruktur eine Rolle, nämlich die Komplexität des Problemfeldes. Damit ist die Menge an Wissen gemeint, die notwendig ist, um ein Problem in diesem Bereich zu lösen. Sie darf nicht zu klein sein, da ansonsten das Problem auch auf konventionelle Weise

bzw. mit wenigen Heuristiken lösbar ist und sich die Erstellung eines Expertensystems nicht lohnt. Anderer seits darf sie nicht zu groß sein, da sonst ein Expertensystem mit einer akzeptablen Fehlerquote nicht mit vertretbarem Aufwand erstellt werden kann.

Die Komplexität eines Problembereichs hat zwei Dimensionen. Unter der Weite eines Problembereichs kann man die Menge der Probleme verstehen, die zu dem Problembereich gezählt wird. So ist das Problemfeld "Auswahl eines Wertpapieres" vergleichsweise weit definiert. Die Teilprobleme "Auswahl einer Aktie" bzw. "Auswahl einer Aktie der High-Tech-Branche" sind demgegenüber zunehmend enger gewählt. Die Weite eines Problembereichs wird durch seine Abgrenzung definiert, sie ist also vor der Konstruktion eines Expertensystems wählbar.

Die richtige Eingrenzung des Problembereichs ist ein wesentlicher Faktor für den Erfolg einer Expertensystem-Anwendung. Ist sie zu eng, wird das resultierende System später dem Anwender kaum einen Nutzen bringen. Ist sie zu weit, wird eine spätere Reduzierung des Anspruches an das System unvermeidlich sein, was zu Enttäuschung und Frustration der Verantwortlichen bis hin zum Abbruch des Projektes führen kann. Gerade hier liegt die besondere Gefahr einer zu voreiligen und optimistischen Einschätzung der Möglichkeiten der Expertensystem-Technologie, da zu hohe Erwartungen an die Leistungsfähigkeit von Expertensystemen gestellt bzw. geweckt werden könnten. In jedem Fall ist eine anfangs zu enge Eingrenzung weniger gravierend, da eine spätere Erweiterung des Problembereichs leicht möglich ist.

Unter der Tiefe eines Problembereichs soll die Menge an Wissen verstanden werden, die zur Lösung des kleinstmöglichen Teilproblems des Problembereichs benötigt wird. Dies wäre im Beispiel der Anlageberatung z.B. die

Frage: "Soll die Aktie der XY-AG gekauft werden?". Die Tiefe des Problembereichs wird durch die Zahl der bei der Lösungsfindung zu berücksichtigenden Faktoren sowie durch die Problemstruktur definiert. Dabei gilt, daß die Problemtiefe umso größer wird, je mehr Defekte das untersuchte Problem aufweist, da Mängel in der Problemstruktur in der Regel durch eine Erweiterung der Wissensbasis kompensiert werden müssen.

Die Gesamt-Komplexität eines Problembereichs wird durch die Kombination beider Dimensionen bestimmt. Das Problem der Wertpapier-Anlageberatung ist z.B. relativ weit definiert, da eine große Menge möglicher Wertpapiere zu berücksichtigen ist, während eine verhältnismäßig geringe Problemtiefe vorliegt, da jedes Wertpapier nur wenige relevante Merkmale (Rendite, Risiko, Mindest-Anlagedauer sowie einige Nebenbedingungen wie Mindest-Anlagebetrag, steuerliche Aspekte etc.) aufweist.

Demgegenüber besitzt die Frage, ob eine bestimmte Investition durchgeführt werden soll oder nicht, eine geringe Problemweite, da nur eine Alternative geprüft wird. Die Zahl der zur Prüfung dieser Frage zu berücksichtigenden Faktoren und damit die Problemtiefe ist jedoch in der Regel sehr groß. Hinzu kommt, daß eine Investitionsentscheidung in der Praxis meist erhebliche Strukturdefekte aufweist, da z.B. viele relevante Größen nicht quantifiziert werden können bzw. zum Teil widersprüchliche Zielsetzungen (Gewinn, Risiko) vorliegen.

Zur Beurteilung der Frage, ob ein gegebenes Problem hinsichtlich seiner Komplexität von einem Expertensystem bewältigt werden kann, kann folgende Faustregel herangezogen werden: "Ein Expertensystem kann solche Probleme lösen, die ein Experte am Telefon innerhalb von zwanzig Minuten bearbeiten kann."[1] Die Einschränkung auf das Telefon soll andeuten, daß in der Regel nur verbale Kommunikation mit dem Benutzer möglich ist und nicht etwa die Zuhilfenahme von Schaubildern, Plänen etc.

Zum Bereich der Problemkomplexität gehört auch die Frage nach der Bedeutung von Allgemeinwissen, die in Abschnitt 3.2.3 behandelt wurde. Erfordert ein Problem Allgemeinwissen, so ist seine Komplexität viel zu hoch, da praktisch das gesamte Alltagswissen eines Menschen erforderlich ist, um es zu lösen, bzw. weil der relevante Teilbereich im voraus nicht abgegrenzt werden kann. In diesem Fall muß ein Teilbereich des Problems gewählt werden, der allein mit fachspezifischem Wissen gelöst werden kann.

Zusammenfassend kann gesagt werden, daß ein Problem dann als aussichtsreicher Kandidat für ein Expertensystem angesehen werden kann, wenn

- es ein Routineproblem ist, das von einem Experten in weniger als einer halben Stunde gelöst werden kann und
- keine Kreativität oder Allgemeinwissen zur Lösung erforderlich sind und
- die Lösung des Problems in der Praxis durch Heuristiken erfolgt, weil konventionelle, exakte Methoden nicht verfügbar oder zu aufwendig sind und
- das Problem von erfahrenen Experten wesentlich besser gelöst wird als von Laien.

1) O'Keefe, Brenton, Ball (66), S.657.

3.3.2 Problemkategorien

In der Literatur finden sich verschiedene Ansätze, um die Problemkategorien, die von Expertensystemen gelöst werden können, zu systematisieren[1]. Diese Systematiken beschränken sich jedoch in der Regel darauf, vorhandene Expertensysteme anhand ihres Einsatzbereiches und ihrer Problemlösungsmethoden in mehr oder weniger homogenen Clustern zu aggregieren.

Demgegenüber soll hier der Versuch gemacht werden, die Probleme anhand der Dimensionen "Lösungsraum" und "Zeitbezug" aufzugliedern. Die sich ergebende Vier-Felder-Matrix ist in Abb. 3.3 wiedergegeben. Jedem der Felder lassen sich typische Lösungsschwierigkeiten und -methoden zuordnen.

Lösungsraum \ Zeitbezug	Statisch	Dynamisch
Explizite Beschreibung	Identifikation	Interpretation
Implizite Beschreibung	Design	Planung des Zeitablaufs

Abb. 3.3: Problemkategorien

1) Vgl. z.B. Mertens, Allgeyer, Däs (53), S.908 f., Puppe (68), S.3, Waterman (91), S.32 f.

Der Lösungsbereich eines Problems kann sowohl explizit als auch implizit beschrieben werden. Bei der expliziten Beschreibung sind alle möglichen Lösungen bereits vollständig in der Wissensbasis enthalten. Das Expertensystem hat die Aufgabe, die den jeweiligen Umständen (fallspezifische Daten) am besten entsprechende Lösung auszuwählen. Als Spezialfall der expliziten Beschreibung des Lösungsraumes kann die Ja/Nein-Entscheidung angesehen werden. Hier wird der Lösungsraum lediglich durch die beiden Elemente "Ja" und "Nein" bzw. "Annahme" und "Ablehnung" gebildet.

Als Beispiel für die explizite Darstellung des Lösungsraumes kann wiederum das Problem der Wertpapier-Anlageberatung angeführt werden. Hier sind alle Elemente des Lösungsraumes, nämlich die einzelnen Wertpapiere, explizit in der Wissensbasis aufgeführt. Das Expertensystem hat nun die Aufgabe, dasjenige Wertpapier auszuwählen, das den Umständen, nämlich den Anforderungen des Anlegers, am besten entspricht.

Bei der impliziten Beschreibung des Lösungsraumes dagegen sind die Elemente des Lösungsraumes nicht in der Wissensbasis enthalten. Sie müssen vielmehr zunächst entsprechend der Beschreibung des Lösungsraumes konstruiert und dann mit den Rahmendaten verglichen werden. Entspricht die konstruierte Lösung den Erfordernissen der Problemstellung, ist das Problem gelöst. Anderenfalls muß die konstruierte Lösung variiert oder verworfen und neu konstruiert werden.

Ein praktisches Beispiel für die implizite Beschreibung des Lösungsraumes ist das Problem der Konfiguration von Computeranlagen[1]. Dabei geht es darum, aus einer Vielzahl von möglichen Komponenten diejenigen auszuwählen,

1) Vgl. das Beispiel XCON in Abschnitt 3.4.1.

die den Anforderungen des Benutzers am besten entsprechen. Dabei sind zahlreiche Restriktionen hinsichtlich der Kombinationsmöglichkeiten zu berücksichtigen: Eine bestimmte Menge an Speicherkapazität erfordert z.B. eine bestimmte Zentraleinheit, die wiederum eine besondere Stromversorgung benötigt usw. Hier ist der Lösungsraum durch die Restriktionen nur implizit beschrieben, es wird nicht jede denkbare Kombination explizit aufgeführt (dies wäre praktisch unmöglich).

Die implizite Darstellung des Lösungsraumes stellt wesentlich höhere Anforderungen an die Inferenzkomponente eines Expertensystems, da die Lösung nun nicht mehr durch reines Suchen in der Wissensbasis ermittelt werden kann. Stattdessen müssen Methoden vorhanden sein, die es erlauben, eine mögliche Lösung aus Einzelelementen zu konstruieren, zu verändern und bereits früher getroffene Entscheidungen wieder zurückzunehmen.

Hinsichtlich des Zeitbezugs können statische und dynamische Probleme unterschieden werden. Bei statischen Problemen spielt der Zeitbezug keine Rolle, d.h. zur Lösung des Problems müssen keinerlei zeitliche Beziehungen berücksichtigt werden. Bei dynamischen Problemen müssen dagegen sich im Zeitverlauf verändernde Größen analysiert oder gestaltet werden.

Während statische Probleme keine besonderen Anforderungen an die Inferenzkomponente stellen, bereitet die Darstellung zeitlicher Beziehungen in der Wissensbasis teilweise erhebliche Schwierigkeiten. Aus diesem Grunde werden bei der Lösung dynamischer Probleme durch Expertensysteme oft konventionelle Methoden hinzugezogen, die das Analysieren dynamischer Systeme erlauben, wie etwa die Simulation. Das Expertensystem dient dann z.B. dazu, das der jeweiligen Problemstellung entsprechende

Simulationsmodell zu konstruieren und die Ergebnisse der Simulation zu interpretieren bzw. zu erklären.

Ein Sonderfall eines dynamischen Problems liegt vor, wenn sich die Problemumgebung, d.h. die für die Lösungsfindung relevanten Informationen, während der Problemlösung verändern können. In diesem Fall muß das Expertensystem in der Lage sein, sehr schnell auf diese Veränderungen zu reagieren. Man spricht auch von sog. Echtzeit-Problemen. Sie stellen besonders hohe Anforderungen an die Rechengeschwindigkeit eines Expertensystems. Ein Beispiel für eine solche Problemstellung ist die automatische Steuerung eines Produktionsprozesses oder einer Maschine durch ein Expertensystem.

Kombiniert man eine explizite Beschreibung des Lösungsraumes mit einem statischen Problemtyp, so erhält man die Problemklasse der Identifikation. Diese Problemklasse hat die einfachste Struktur und bildet die häufigste Anwendungsklasse für Expertensysteme. Im betriebswirtschaftlichen Bereich gehören vor allem Auswahlprobleme wie etwa die Wahl eines geeigneten Standortes, eines Lieferanten oder eines Wertpapieres in diese Klasse. Kennzeichnend ist, daß alle möglichen Alternativen bekannt und explizit in der Wissensbasis formuliert sein müssen, zwischen denen nach bestimmten Kriterien ausgewählt wird.

Identifikationsprobleme bringen aus der Sicht der Expertensystem-Technologie die geringsten Probleme mit sich. Da der Lösungsraum auf eine relativ geringe Zahl von Elementen begrenzt ist, kann in der Regel eine Lösung mit einfachen Suchstrategien gefunden werden. Dabei werden meistens die Merkmale der im Lösungsraum enthaltenen Elemente mit den Anforderungen der jeweiligen Problemsituation verglichen und das Element mit der größten übereinstimmung ausgewählt.

So wählt beispielsweise ein Expertensystem zur Wertpapier-Anlageberatung, das zur Zeit mit der Expertensystem-Shell DONALD konstruiert wird[1], das für einen Kunden am besten geeignete Wertpapier aus, indem es dessen Anforderungsprofil hinsichtlich Risiko, Rendite, Liquidierbarkeit, steuerlichen Aspekten etc. mit den entsprechenden Eigenschaften jedes Wertpapiers vergleicht. Dabei wird das Konzept der Frames[2] benutzt, um die Wissensbasis möglichst effizient formulieren zu können. So ist beispielsweise die Wertpapierklasse "Aktien" für einen risikoscheuen Anleger nicht geeignet. Alle Unterklassen des Frames "Aktien" werden daher in diesem Fall nicht weiter untersucht.

Schwierigkeiten treten vor allem dann auf, wenn ein zielsetzungsdefektes Problem[3] vorliegt, d.h. wenn die Anforderungen des jeweiligen Anwendungsfalls teilweise widersprüchlich sind. So widersprechen sich die Zielsetzungen "Renditemaximierung" und "Risikominimierung" im Fall der Wertpapierauswahl. Es muß ein Weg gefunden werden, um den hier auftretenden Bewertungskonflikt zu lösen bzw. zu umgehen. Dies kann z.B. dadurch erfolgen, daß ein Anforderungsprofil des Benutzers, das die Merkmale "Maximale Rendite" und "Kein Risiko" enthält, vom System abgelehnt wird, so daß der Benutzer sich für eine der beiden Extremlösungen oder einen Kompromiß entscheiden muß.

---

1) Dieses Expertensystem wird vom Institut für Numerische und Instrumentelle Mathematik der Universität Münster unter der Leitung von Prof.Dr. W.Cryer in Zusammenarbeit mit der Volksbank Münster entwickelt. Vgl. Koch, Heuer (41).
2) Vgl. Abschnitt 2.3.3.
3) Vgl. den vorigen Abschnitt.

Geht man von der expliziten zur impliziten Beschreibung des Lösungsraumes bei statischem Problemtyp, so gelangt man zur Problemkategorie der Design- oder Gestaltungsprobleme. Hier muß zunächst eine zulässige Lösung erzeugt werden, die dann wie beim Identifikationsproblem mit den Anforderungen der realen Problemsituation verglichen wird. Eine Lösung besteht zumeist aus einer Kombination von Werten oder einzelnen Elementen, von denen jedes aus einer größeren Zahl von Möglichkeiten ausgewählt werden muß.

Der Lösungsraum wird in der Regel dadurch beschrieben, daß für jedes Element der Lösung die möglichen Ausprägungen definiert werden. Zusätzlich können bestimmte Kombinationen von Merkmalen ausgeschlossen oder vorgeschrieben werden. Da die verbleibende Zahl zulässiger Lösungen meist immer noch sehr groß ist, müssen effiziente Verfahren zur Erzeugung und Bewertung möglicher Lösungen herangezogen werden.

Im betriebswirtschaftlichen Bereich treten Design-Probleme z.B. überall dort auf, wo knappe Ressourcen unter Vernachlässigung zeitlicher Beziehungen auf verschiedene Teilbereiche oder Funktionen aufgeteilt werden müssen. Beispiele sind die Aufteilung eines Werbebudgets auf verschiedene Medien oder die Aufteilung von Vertretern auf verschiedene Produkte oder Kunden.

Ein Problem, das in enger Beziehung zu betriebswirtschaftlichen Fragestellungen steht, ist das der Konfiguration technischer Anlagen. Dabei spielen sowohl technische (welche Kombinationen sind möglich?) als auch betriebswirtschaftliche (welche Kombination ist die kostengünstigste?) Fragestellungen eine Rolle. Das Expertensystem XCON zur Konfiguration von Computeranlagen, das in Abschnitt 3.4.1 näher beschrieben wird,

demonstriert eindrücklich den Nutzen, den Expertensysteme in diesem Bereich bieten können[1].

Die explizite Darstellung des Lösungsraumes unter Einbeziehung des Zeitablaufs führt zur Problemklasse der Interpretation. Hier müssen wiederum alle Lösungsmöglichkeiten explizit in der Wissensbasis enthalten sein. Im Unterschied zur Identifikation werden jedoch auch Zeitbeziehungen des realen Problems berücksichtigt. Dies bedeutet, daß das Expertensystem über Methoden zur Analyse von Zeitreihen oder zur Simulation verfügen muß.

Ein typisches Beispiel für ein Interpretationsproblem im betriebswirtschaftlichen Bereich ist die Bilanz- bzw. GuV-Analyse. Die Expertensysteme "BILEX" und "GUVEX", die am Lehrstuhl für Betriebsinformatik der Universität Erlangen-Nürnberg entwickelt werden[2], versuchen beispielsweise, aus der Entwicklung verschiedener Bilanz- und GuV-Positionen auf die wirtschaftliche Situation eines Unternehmens zu schließen und so den Wirtschaftsprüfer bei seiner Arbeit zu unterstützen. Dabei werden jeweils die GuV bzw. Bilanz des Vor- und des Berichtsjahres miteinander verglichen.

Auch Prognosen gehören in den Bereich der Interpretationsprobleme. So kann das Expertensystem FOUNDER zur Beratung bei Existenzgründungen, das in Kapitel 4 beschrieben wird, als Interpretationssystem aufgefaßt werden, da es eine Prognose der Liquidität des zu gründenden Unternehmens umfaßt. Es benutzt dazu ein Simulationsprogramm, das als MOSES-Modul integriert ist. Der

---

1) Dieses System führt nach Angaben des Anwenders zu jährlichen Kosteneinsparungen in Millionenhöhe. Vgl. Schieferle (75), S.54 sowie Abschnitt 3.4.1.
1) Vgl. Dräger u.a. (19), Büttner u.a. (12).

Lösungsraum umfaßt nur die Aussage "Gründung ist erfolgversprechend" bzw. die Gegenaussage "Gründung ist nicht erfolgversprechend".

Besondere Schwierigkeiten treten bei Interpretationsproblemen dadurch auf, daß Zeitabläufe vom Expertensystem analysiert werden müssen. Dies kann, wie im Beispiel des Systems POUNDER, durch Unterprogramme in konventioneller Programmiertechnik, die in das System integriert werden, geschehen.

Expertensysteme zur Lösung von Interpretationsproblemen sind oft auch schon deswegen schwieriger zu erstellen, weil menschliche Experten hier an ihre Grenzen stoßen. Dies zeigt das Beispiel der Bilanzanalyse deutlich: Zahlreiche spektakuläre Pleiten konnten selbst von den besten Bilanzanalytikern nicht vorhergesehen werden. Der Grund liegt darin, daß die Bilanz bzw. GuV nicht alle für die Beurteilung eines Unternehmens notwendigen Informationen enthält und zudem durch Bilanzmanipulation ein bewußt falsches Bild der Unternehmenssituation dargestellt wird.

Verallgemeinert man dies, so kann gesagt werden, daß bei vielen Interpretationsproblemen einerseits nicht alle relevanten Größen beurteilt werden können, weil sie dem Analytiker nicht zugänglich sind. Zum anderen werden oft die Zusammenhänge zwischen beobachteten Tatbeständen und den dahinter stehenden Ursachen nicht hinreichend verstanden, d.h. es handelt sich um wirkungsdefekte Probleme[1].

Die komplexeste und schwierigste Problemklasse ist die Planung des Zeitablaufs. Die Lösung muß zunächst

1) Vgl. den vorigen Abschnitt.

durch Aneinanderreihen einzelner Lösungselemente (z.B. Arbeitsschritte) erzeugt werden, wobei zeitliche Restriktionen zu beachten sind (bei der Werkstattfertigung kann z.B. ein Werkstück nicht gleichzeitig auf zwei Maschinen bearbeitet werden).

Die Planung des Zeitablaufs[1] ist ein zentrales betriebswirtschaftliches Problem, das in allen betrieblichen Funktionsbereichen eine Rolle spielt. Dabei geht es meist darum, knappe Mittel (z.B. Personal oder Maschinen) auf verschiedene Aufgaben (z.B. Tätigkeiten oder Aufträge) zu verteilen, wobei im Unterschied zum Design-Problem zeitliche Abläufe eine Rolle spielen. Beispiele sind die Personaleinsatzplanung, die Maschinenbelegungsplanung oder die Tourenplanung.

Expertensysteme können möglicherweise gerade auf diesem Gebiet einen wertvollen Beitrag zum betriebswirtschaftlichen Instrumentarium leisten, da es sich bei diesen Planungstätigkeiten oft um Routineprobleme handelt, die von Experten anhand von Wissen über Restriktionen und Heuristiken gelöst werden und die mit konventionellen mathematischen Methoden meist nicht lösbar sind.

Allerdings stellt diese Problemklasse auch die größten Anforderungen an die Inferenzkomponente und die Methoden zur Wissensdarstellung innerhalb eines Expertensystems. Zum einen ist ein Plan in der Regel ein sehr komplexes Gebilde, das eine entsprechend komplexe Wissensdarstellung erfordert, die auch die zahlreichen Restriktionen und Abhängigkeiten berücksichtigt. Zum an

---

1) Unter "Planung des Zeitablaufs" soll hier die Erstellung einer zeitlich geordneten Folge von Handlungsanweisungen verstanden werden, die einen gegebenen Ausgangszustand in einen gewünschten Endzustand überführt.

deren ist der Lösungsraum meist so groß, daß effiziente Verfahren zur Erzeugung bzw. Verbesserung von Lösungen entwickelt werden müssen.

Aus diesem Grunde sind Expertensysteme zur Planung des Zeitablaufs noch vergleichsweise selten. Ein herausragendes Beispiel bildet das System ISIS[1] zur Ablaufplanung bei Werkstattfertigung. Die Komplexität des zugrundeliegenden Problems wird durch die Vielzahl der zu berücksichtigenden Faktoren verdeutlicht:

"Die Annehmbarkeit eines spezifischen Ablaufplans hängt von so vielfältigen und widersprüchlichen Faktoren wie Fertigstellungsterminen, Kosten-Restriktionen, Auslastungsgraden, Eignung und Substituierbarkeit von Maschinen, alternativen Produktionsprozessen, Auftrags-Charakteristika sowie Bedarf und Verfügbarkeit von Rohstoffen ab."[2]

Um diesen vielfältigen Restriktionen gerecht zu werden, kombiniert das System eine Vielzahl von Repräsentationsformen und Lösungsmethoden. Ein Kernprinzip des Systems ist die Methode der "Constraint-Propagation"[3]. Dabei werden zuerst diejenigen Arbeitsschritte geplant, für die die stärksten bzw. meisten Restriktionen bestehen. Ein auf diese Weise erstellter Ausgangsplan wird schrittweise verbessert, indem durch Umstellen von Arbeitsschritten eine zulässige bzw. günstigere Lösung zu erreichen versucht wird. Dabei müssen die vielfältigen, vernetzten Auswirkungen einer Umstellung von vornherein berücksichtigt werden. Dies geschieht durch spezielle

1) Vgl. Fox, Smith (23).
2) Ebenda, S.25. übersetzung durch den Verfasser.
3) Vgl. z.B. Stefik (84).

Elemente der Wissensbasis, in denen die Interdependenzen der verschiedenen Restriktionen gespeichert sind.

Ein weiteres Anwendungsbeispiel für ein Expertensystem zur Erstellung von Plänen ist das vom Battelle-Institut entwickelte System GUMMEX zur Erstellung von Arbeitsplänen bei der Herstellung von Gummi-Membranen[1]. Das zugrundeliegende Problem weist jedoch eine wesentlich einfachere Struktur als die Ablaufplanung bei Werkstattfertigung auf, da sich die Herstellung von Gummi-Membranen in voneinander unabhängige Teilschritte zerlegen läßt. Eine Maschinenbelegungsplanung sieht das System nicht vor, es werden lediglich die einzelnen Arbeitsschritte zur Herstellung des Produktes spezifiziert.

Abschließend soll darauf hingewiesen werden, daß Anwendungsprobleme für Expertensysteme auch nach der Zielsetzung unterschieden werden können, und zwar in Expertensysteme zur Entscheidungsfindung und solche zur Wissensvermittlung. Letztere können, abhängig vom Stoffgebiet, Merkmale aller genannten Problembereiche aufweisen. Hinzu kommt bei diesen Systemen, daß sie in der Regel ein Wissens-Modell des Benutzers bilden, anhand dessen sie die Wissensvermittlung vor allem auf Kenntnislücken konzentrieren können[2].

Expertensysteme zur Wissensvermittlung sind nicht Gegenstand dieser Arbeit, da sie eigentlich keine betriebswirtschaftlichen, sondern pädagogische Aufgaben lösen. Dennoch könnten solche Systeme in der Betriebs

---

1) Vgl. Iudica (38), Trum (89).
2) Vgl. Waterman (91), S.37 f.

wirtschaftslehre, vor allem im Ausbildungsbereich, große Bedeutung erlangen. Da diese Systeme nicht zur Vorbereitung von Entscheidungen dienen, treffen einige der in Abschnitt 3.2 genannten Einschränkungen auf sie nicht zu[1].

Die hier vorgestellte Systematik ist weder allumfassend noch völlig überschneidungsfrei. Sie bietet jedoch die Möglichkeit, die meisten praktischen Probleme einzuordnen und die technischen Schwierigkeiten bei der Realisierung eines Expertensystems abzuschätzen. Dabei kann als Faustregel gelten, daß die Realisierung um so schwieriger wird, je weiter rechts und je weiter unten das Problem in der Matrix eingeordnet wird. Damit ist jedoch noch nichts über die Schwierigkeiten der Wissensakquisition und die generelle Lösungsmöglichkeit mit Hilfe der Expertensystem-Technologie gesagt.

---

1) Dies betrifft vor allem die Verantwortlichkeits-Problematik. Damit fällt die Forderung weg, der Benutzer müsse die Schlußfolgerungen des Expertensystems stets nachvollziehen können. Eine derartige Voraussetzung an ein System zur Ausbildung zu stellen wäre ohnehin widersinnig.

3.3.3 Die Vorteilhaftigkeit von Expertensystemen

Zur Bestimmung der betriebswirtschaftlichen Vorteilhaftigkeit der Konstruktion eines Expertensystems sind die Erträge in Form von Kosteneinsparungen den durch die Konstruktion und den laufenden Betrieb entstehenden zusätzlichen Kosten gegenüberzustellen[1]. Dazu könnte man folgende Rechnung aufmachen:

```
    Entwicklungskosten durch den
    Knowledge Engineer bzw. eine
    Beratungsfirma
  + Kosten für Hard- und Software
  + Kosten für Freistellung
    des/der Experten
-----------------------------------
  = Erstellungskosten
  + laufende Kosten für Wartung
    und Bedienung
  + Kosten für Fehlentscheidungen
    aufgrund von Fehlern des Systems
-----------------------------------
  = Kosten des Expertensystems


    Einsparungen durch verbesserte
    Produktivität der Systembenutzer
  + Einsparungen durch Entlastung
    des Experten
  + Einsparungen durch vermiedene
    Fehlentscheidungen aufgrund der
    Systemleistung
-----------------------------------
  = Einsparungen durch das Experten-
    system
./. Kosten des Expertensystems
-----------------------------------
  = Vorteilhaftigkeit des Experten-
    systems
```

Schema 3.1: Vorteilhaftigkeit von Expertensystemen

---

1) von Zelewski unternimmt in (99), S.1001 ff. eine detaillierte Kosten/Nutzen-Analyse des Einsatzes von "Intelligenten Automaten" (zu denen er auch Expertensysteme zählt).

Die einzelnen, in Schema 3.1 aufgeführten Größen dürften jedoch nur sehr schwer zu quantifizieren sein. Zum einen ist nämlich eine Prognose der Entwicklungsdauer eines Expertensystems schon deswegen praktisch unmöglich, weil es sich dabei um einen evolutorischen Prozeß[1] und nicht, wie bei EDV-Anwendungen normalerweise üblich, um ein Projekt mit klar definierten Arbeitsschritten und Fertigstellungsterminen handelt.

Zum anderen ist in aller Regel im voraus kaum abzusehen, ob und inwieweit ein Expertensystem im betrachteten Bereich zu einer echten Produktivitätssteigerung führt. Aufgrund der Tatsache, daß die Leistungsfähigkeit eines Expertensystems vom Umfang und der Qualität der Wissensbasis abhängt, sind Entwicklungsdauer und Leistungsvermögen miteinander verknüpfte Entscheidungsvariablen, so daß die meisten in Schema 3.1 genannten Größen keine Rahmendaten sind, sondern in großem Umfang von den Entscheidungsträgern beeinflußt werden können. Das daraus resultierende Planungsproblem dürfte aufgrund der zahlreichen Unsicherheiten kaum mit konventionellen Planungsmethoden lösbar sein[2].

Hinzu kommt, daß im Einzelfall weitere Vorteile durch die Konstruktion des Expertensystems entstehen können, die ebenfalls kaum zu quantifizieren sind[3]. Dazu gehört z.B. der positive Effekt, der durch die Offenlegung und Objektivierung des Expertenwissens entsteht, ebenso wie eine Steigerung der Flexibilität durch kürzere Entscheidungsdauern. Weitere positive Wirkungen können sich durch eine bessere Wissensvermittlung an

1) Vgl. Abschnitt 3.1.1.
2) Es handelt sich um ein wirkungsdefektes Problem, da die Zusammenhänge zwischen der Entwicklungsdauer und der Leistungsfähigkeit des Systems nicht bekannt sind. Vgl. dazu Abschnitt 3.3.1.
3) Vgl. Mertens, Allgeyer (52), S.706 f.

Nachwuchs-Experten, positive Ausstrahlungen auf das Unternehmens-Image oder Kosten- bzw. Informationsvorteile gegenüber der Konkurrenz ergeben.

Da eine Entscheidung über die Anwendung der Expertensystem Technologie aus den gezeigten Gründen nicht auf der Basis einer klassischen Kosten-Nutzen-Analyse erfolgen kann, müssen Heuristiken zur Lösung dieses Problems herangezogen werden. So könnten etwa Indizien herangezogen werden, die für die Anwendung im betrachteten Bereich sprechen[1]. Solche Indizien sind z.B. gegeben,

- wenn die vorhandenen Experten überlastet sind und dadurch häufiger Verzögerungen in der Problembearbeitung entstehen;
- wenn die Gefahr besteht, daß für das Unternehmen bedeutendes Know How verlorgengeht, etwa weil ein Experte in den Ruhestand geht;
- wenn häufig hohe Kosten auftreten, weil gravierende Fehlentscheidungen getroffen werden, die durch mehr Wissen hätten vermieden werden können;
- wenn schnell verfügbares Wissen, z.B. in der Kundenberatung, für das Unternehmen ein wesentlicher Erfolgsfaktor ist.

Weiterhin können bereits existierende Expertensystem-Anwendungen in demselben oder einem ähnlichen Bereich einen Hinweis dafür geben, daß sich eine Entwicklung lohnt. Mertens, Allgeyer, Däs geben in (53) eine Übersicht über betriebliche Expertensystem-Entwicklungen im deutschsprachigen Raum (Stand: Frühjahr 1986), die eine sehr gute Ausgangsbasis für derartige Hinweise bietet.

1) Vgl. dazu auch Waterman (91), S.130 f.

Angesichts der Tatsache, daß die Entwicklung eines Expertensystems erhebliche Investitionen erfordert und personelle Kapazitäten auf Jahre hinaus bindet, muß die Entscheidung über eine Anwendung zumindest zum gegenwärtigen Zeitpunkt eindeutig als strategische Entscheidung aufgefaßt werden.

Dafür spricht auch, daß Expertensysteme einen wesentlichen Einfluß auf die strategische Wettbewerbsposition eines Unternehmens haben können, weil erhebliche Kosten oder Flexibilitätsvorteile gegenüber der Konkurrenz entstehen können. So kann z.B. die Firma Digital Equipment heute jede einzelne Rechenanlage nach Kundenwünschen konfigurieren und somit den Erfordernissen des Kunden optimal anpassen. Dies wäre ohne die Entwicklung des Expertensystems XCON sehr wahrscheinlich nicht möglich gewesen[1]. Expertensysteme können somit als strategische Waffe eingesetzt werden[2].

Trotz des damit verbundenen hohen Aufwandes ist daher eine eingehende Beschäftigung mit der Anwendung von Expertensystemen schon deswegen empfehlenswert, weil dadurch Erfahrungen mit einer Technologie gewonnen werden, die möglicherweise die Wettbewerbssituation der Unternehmen in den kommenden Jahren erheblich beeinflussen wird. Hinzu kommen die bereits erwähnten Nebeneffekte, die den Aufwand teilweise auch dann rechtfertigen können, wenn letztendlich kein im praktischen Betrieb eingesetztes Expertensystem erstellt werden kann.

---

1) Vgl. Schieferle (75), S.54.
2) Vgl. Mertens, Plattfaut (54), insb. S.16 f.

3.4 Anwendungsperspektiven im betriebswirtschaftlichen Bereich

3.4.1 Betriebswirtschaftliche Anwendungsmöglichkeiten: Das Beispiel der Digital Equipment Corporation

In den vorhergehenden Abschnitten wurde gezeigt, welche Arten von Problemen mit Expertensystemen sinnvoll gelöst werden können. In diesem Abschnitt sollen nun konkrete Anwendungsmöglichkeiten aufgezeigt werden. Es kann jedoch nicht Aufgabe dieser Arbeit sein, alle denkbaren Problembereiche aufzuzählen, für die ein Expertensystem erstellt werden könnte.

Im folgenden soll daher am Beispiel der Firma Digital Equipment Corporation (im folgenden DEC genannt), die als Pionier im Bereich der industriellen Nutzung von Expertensystemen gilt[1], dargestellt werden, daß Expertensysteme nicht nur für einzelne betriebswirtschaftliche Probleme erstellt werden können, sondern daß sie zu einem Netzwerk von Systemen zusammenwachsen können, die auf gemeinsame Datenbestände zugreifen und miteinander kommunizieren.

Die breite und umfassende Nutzung von Expertensystemen bei DEC belegt zum einen, daß diese Technologie in sehr vielen unterschiedlichen, vielleicht sogar in allen betrieblichen Funktionsbereichen sinnvoll eingesetzt werden kann[2]. Zum anderen zeigt sie, daß sich die Beschäftigung mit der Expertensystem-Technologie vor allem aus langfristig strategischen Überlegungen heraus lohnt, da der Nutzeffekt einer Anwendung weit über den

---

1) Vgl. Mertens, Allgeyer, Däs (53), S.913.
2) Dies belegt auch die Aufstellung von Expertensystem Entwicklungen bei Mertens, Allgeyer, Däs (53), S.910 ff., die keinen wesentlichen betrieblichen Funktionsbereich ausläßt.

ursprünglichen Problembereich hinaus gehen und eine übertragung der gewonnenen Erfahrungen auf andere betriebliche Teilbereiche möglich sein kann.

Mitte der siebziger Jahre wurde bei DEC die strategische Entscheidung getroffen, die hergestellten Computeranlagen speziell nach Kundenwünschen und -bedürfnissen zu konfigurieren[1]. Dies bedeutete, daß aus einer Menge von mehreren tausend Einzelkomponenten bei jedem Auftrag einige hundert in einer neuen Kombination zusammengestellt werden mußten. Da vielfältige Abhängigkeiten und Restriktionen bezüglich der Kombinationen von Komponenten zu beachten sind, wären zur Erfüllung dieser Aufgabe 1983 etwa 1000 Konfigurationsspezialisten notwendig gewesen[2].

Da dies einen erheblichen Kostenfaktor für das Unternehmen bedeutete, wurden mehrere Versuche unternommen, die Konfigurationsaufgabe mit Hilfe konventioneller Software-Ansätze zu automatisieren. Diese Versuche scheiterten im wesentlichen, weil das zugrundeliegende Problem zu komplex war, um es in einem analytischen Modell abbilden zu können, und weil beinahe täglich neue Komponenten entwickelt wurden, die bei den Konfigurationen zu berücksichtigen waren[3]. Die konventionellen Ansätze erwiesen sich als zu starr, um dieser sich rasch verändernden Problemsituation gerecht werden zu können.

Schließlich wandte man sich im Jahr 1978 an John McDermott von der Carnegie Mellon Universität, der sich mit der Anwendung von Expertensystemen beschäftigte. Nachdem dieser sich in das Problem der Konfigurierung von

1) Vgl. Schieferle (75), S.54.
2) Vgl. ebenda.
3) Vgl. Harmon, King (33), S.176.

Computeranlagen eingearbeitet hatte, erstellte er auf der Basis zweier Konfigurationshandbücher von DEC innerhalb weniger Monate einen ersten Prototyp, wobei er die ebenfalls an der Carnegie Mellon Universität entwickelte, regelbasierte Knowledge-Engineering-Sprache OPS4 verwendete[1]. Bei der Vorführung konnte das von McDermott R1[2] genannte System bereits einfache Konfigurierungsaufgaben lösen und so die erhebliche Skepsis des Managements bezüglich der Anwendbarkeit von Expertensystemen ausräumen.

Man machte sich nun daran, ein System zu entwickeln, das im praktischen Betrieb eingesetzt werden konnte. Das System sollte Computersysteme des Typs VAX-11/780 in 75% der Fälle richtig konfigurieren können. Dieses Ziel wurde bereits im September 1979 erreicht, weniger als ein Jahr nach Beginn der ersten Vorüberlegungen. Das resultierende System hatte einen Umfang von 750 Regeln[3].

Im Laufe der Zeit wurde das System in zwei Richtungen weiterentwickelt: zum einen wurde die Breite des Problembereichs vergößert, indem auch andere Produktlinien von DEC einbezogen wurden, zum anderen wurde die Tiefe der Wissensbasis vergößert, indem auch selten zu beachtende Konfigurationsregeln aufgenommen wurden[4], d.h. die Zahl der Fehlkonfigurationen wurde gesenkt. 1986 konnte das inzwischen XCON genannte System alle von DEC hergestellten Computeranlagen konfigurieren und machte dabei nur noch in 1% der Fälle einen Fehler[5].

---

1) Vgl. Harmon, King (33), S.177.
2) McDermott bemerkt zur Namenswahl in (47), S.39, ironisch: "Four years ago I couldn´t even say ´Knowledge Engineer´, now I ...".
3) Vgl. Harmon, King (33), S.177.
4) Somit wurde die Problemkomplexität schrittweise erhöht; vgl. Abschnitt 3.3.1.
5) Vgl. Schieferle (75), S.55 f.

Angeregt durch den Erfolg von XCON suchte man bei DEC nach weiteren Anwendungsfeldern für die neue Technologie. Man entwickelte zunächst, ebenfalls in Zusammenarbeit mit der Carnegie Mellon Universität, das System XSEL zur Unterstützung der Kundenberatung. Dabei geht es darum, in Abhängigkeit von den Kundenbefürfnissen eine Liste von Basiskomponenten zu erstellen, die als Eingabe für das System XCON dient[1].

Gibt der Kunde beispielsweise lediglich die Information "120 alphanumerische Terminals gewünscht" an, so ermittelt XSEL die für diese Anforderungen benötigte Zentraleinheit, ausreichende Haupt- und Massenspeicherkapazitäten etc. Gegebenenfalls benötigte weitere Informationen werden im Dialog mit dem Kunden erfragt. XSEL kann als eine Art "intelligente Benutzerschnittstelle" für XCON angesehen werden, die gleichzeitig eine erhebliche Erweiterung des Problemfeldes darstellt[2].

Im Laufe der Jahre entstand ein Netz aus Expertensystem-Anwendungen, das sich über viele betriebliche Funktionsbereiche ausbreitete. Die wichtigsten Komponenten sind in Abb. 3.4 schematisch dargestellt[3]. Insgesamt waren 1986 bei DEC 36 verschiedene Expertensysteme im Einsatz, zahlreiche weitere befanden sich in der Entwicklung[4]. Die Erfahrungen in der Entwicklung und Anwendung von XCON ermöglichten es, Prototyp-Systeme für andere Bereiche in kurzer Zeit zu erstellen und "ein Gespür für die richtigen Anwendungen" zu entwikkeln[5].

---

1) Vgl. McDermott (48).
2) Vgl. ebenda, S.329.
3) Vereinfachte Darstellung in Anlehnung an Schieferle (75), S.58.
4) Vgl. ebenda, S.54 und S.59.
5) Vgl. ebenda, S.55.

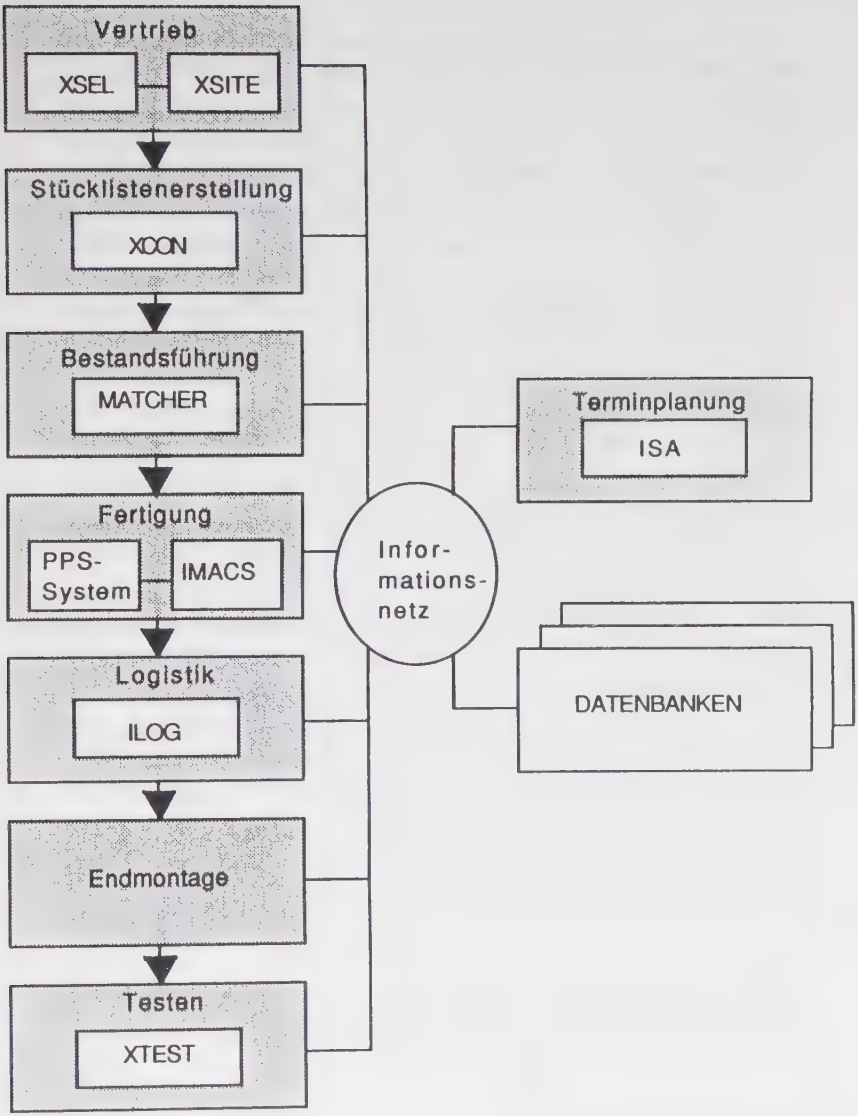


Abb. 3.4: Ausgewählte Expertensysteme bei DEC

Das System XSITE arbeitet mit XSEL und XCON zusammen und ermittelt auf der Basis der räumlichen Verhältnisse des Kunden zusätzliche Angaben für die Konfiguration wie Schranktypen, Kabellängen etc. sowie Hinweise für evtl. notwendige bauliche Maßnahmen. Dies ist insofern bemerkenswert, als die Analyse räumlicher Beziehungen zusätzliche Schwierigkeiten für die Repräsentation des Wissens mit sich bringt. Es demonstriert die Vielseitigkeit und Leistungsfähigkeit der verwendeten Knowledge-Engineering-Sprache OPS5[1].

Betriebswirtschaftlich von besonderem Interesse sind die Systeme "Matcher", IMACS, ILOG und ISA. Die Aufgabe des "Matchers" ist es, die für einen Kunden erstellte Konfiguration mit am Lager befindlichen Computersystemen oder Teilkomponenten zu vergleichen. Dabei wird nach "echten" Ähnlichkeiten zwischen der geforderten und den vorhandenen Konfigurationen gesucht und der Aufwand einer Umrüstung des ähnlichsten vorhandenen Systems abgeschätzt. "Echte" Ähnlichkeit bedeutet dabei, daß nicht allein die Zahl der Abweichungen in der Stückliste als Ähnlichkeitskriterium herangezogen wird. Vielmehr wird auch berücksichtigt, daß z.B. das Erfordernis eines zusätzlichen Stromanschlusses u.U. weitreichende Änderungen nach sich ziehen kann, weil nun das Netzteil nicht mehr ausreicht, ein größeres Netzteil aber nicht in das vorhandene Gehäuse paßt usw[2].

---

1) OPS5 ist eine Weiterentwicklung der von McDermott zu Beginn verwendeten Knowledge-Engineering-Sprache OPS4. Vgl. Harmon, King (33), S.177 und Schieferle (75), S.56 f.
2) Vgl. Schieferle (75), S.57 f.

IMACS, das "Intelligent Manufacturing Control System", analysiert Abweichungen des tatsächlichen Fertigungsprozesses, der mit Hilfe eines konventionellen Produktions-Planungs- und Steuerungssystems (PPS) gesteuert wird, vom Plan, schlägt Maßnahmen zur Behebung der Störung vor und gibt eine Prognose der Auswirkungen dieser Maßnahmen auf den gesamten Fertigungsverlauf[1].

Das System ILOG ("Intelligent Logistic") schließlich steuert den Transport und die Zwischenlagerung der in verschiedenen Werken hergestellten bzw. benötigten Komponenten, wobei die Beanspruchung von Fertigungskapazitäten sowie Transport- und Lagerkosten minimiert werden sollen[2].

ISA, der "Intelligent Scheduling Assistant", hilft bei der Ermittlung von Lieferterminen für die bestellten Computersysteme, wobei die Konfiguration mit dem vorhandenen Lagerbestand verglichen wird. Umgekehrt kann ISA Liefertermin-Wünsche des Kunden vorgeben und dafür benötigte Komponenten im Lager reservieren[3].

XTEST schließlich ermittelt für jede Konfiguration spezifische Testverfahren, die bei der Endmontage eingesetzt werden. Dabei werden speziell solche Eigenschaften des Systems getestet, die aufgrund bestimmter Konfigurationen und unter bestimmten Spitzenbelastungen zu Störungen führen können. Auf diese Weise können individuelle Schwachpunkte des Systems gezielt entdeckt und behoben werden[4].

Weitere, in der Literatur[5] nicht näher beschriebene Systeme koordinieren z.B. die Durchführung von tech-

1) Vgl. Schieferle (75), S.58.
2) Vgl. ebenda, S.58 f.
3) Vgl. ebenda, S.58.
4) Vgl. ebenda, S.57.
5) Vgl. ebenda, S.59.

nischen Änderungen an Teilen, so daß alle betroffenen Stellen gleichzeitig von der Änderung erfahren, die nötigen Teile dort in ausreichenden Mengen vorhanden sind und Abstimmungsverluste vermieden werden.

Das Beispiel DEC läßt einige, für die Beurteilung betriebswirtschaftlicher Anwendungsmöglichkeiten für Expertensysteme interessante Schlußfolgerungen zu:

1) Es hat sich gezeigt, daß die frühzeitige Beschäftigung mit der Expertensystem-Technologie der Firma DEC einen wesentlichen strategischen Wettbewerbsvorteil[1] verschafft hat, da eine kundenspezifische Konfigurierung ohne XCON nicht möglich oder zu teuer gewesen wäre[2].

2) Die Zahl der inzwischen bei DEC installierten Systeme zeigt, daß die Entscheidung über die Konstruktion von XCON sehr viel weiter reichende Konsequenzen für das Unternehmen hatte als wohl zunächst vermutet[3]. Auch aus diesem Grunde kann die Beschäftigung mit der Expertensystem-Technologie als strategische Entscheidung angesehen werden, die nicht nur an kurzfristigen Vorteilhaftigkeitskriterien gemessen werden sollte.

3) Die Erfahrung bei der Konstruktion eines Expertensystems in einem Bereich erleichtert die übertragung auf andere, auch sehr unterschiedliche Problembereiche erheblich[4].

4) Expertensysteme können zu Netzwerken zusammenwachsen, in denen untereinander Daten ausgetauscht und gemeinsame betriebliche Datenbanken genutzt werden.

---

1) Vgl. dazu Mertens, Plattfaut (54).
2) Vgl. Schieferle (75), S.54.
3) Schieferle formuliert in (75), S.59, daß "... auch bei Expertensystemen der Appetit beim Essen kommt."
4) Vgl. ebenda, S.55.

5) Die schrittweise Vorgehensweise des Knowledge Engineering ist ein wesentlicher Grund für den Erfolg der Expertensystem-Anwendungen. Dies nicht nur, weil die leichte Erweiterungsfähigkeit der Wissensbasis eine Lösung des Konfigurierungsproblems überhaupt erst ermöglicht hat[1], sondern auch, weil dadurch eine schrittweise Eingliederung in die bisherigen betrieblichen Strukturen ermöglicht wird, ohne daß gravierende Umwälzungen bzw. Reorganisationen notwendig sind, wie dies bei der Einführung konventioneller EDV oft erforderlich ist.

6) Wesentlich für den Erfolg des XCON-Projektes war die Beschränkung auf zunächst ein relativ eng begrenztes Problem, nämlich die Konfiguration von Anlagen des Typs VAX-11/780. Hätte man beispielsweise von vornherein versucht, Anlagen des komplizierteren Typs PDP-11 zu konfigurieren, so wäre das Projekt nach McDermotts rückblickender Beurteilung vermutlich gescheitert, da die Problemkomplexität zu hoch gewesen wäre[2]. Das Beispiel DEC zeigt, daß eine anfängliche Beschränkung des Problembereichs später leicht aufgehoben werden kann.

Wie bei jedem Beispiel ist natürlich auch im vorliegenden Fall zu fragen, inwieweit eine übertragung der Beobachtungen auf andere Betriebe, Branchen oder Problembereiche zulässig ist. Dabei dürften drei Aspekte eine Rolle spielen: zum einen ist DEC ein innovativer Computer-Hersteller und insofern neuen Technologien gegenüber vermutlich aufgeschlossener als andere Unternehmen, zudem verfügt man über mit Methoden der Programmierung und Softwaretechnik vertraute Fachleute.

---

1) Vgl. Harmon, King (33), S.176.
2) Vgl. ebenda, S.177.

Zum anderen ist unklar, inwieweit die interne Organisationsstruktur der Firma eine Einführung von Expertensystemen begünstigt hat. Möglicherweise ist hier ein Unternehmen wie DEC in einer Wachstumsbranche den Unternehmen mit eher althergebrachten Strukturen gegenüber im Vorteil. Schließlich ist vielleicht das Konfigurationsproblem für die Entwicklung eines Expertensystems besonders geeignet.

Trotzdem zeigt das Beispiel DEC nach Meinung des Verfassers, daß Expertensysteme im betriebswirtschaftlichen Bereich nicht nur einsetzbar sind, sondern eine erhebliche wirtschaftliche und strategische Bedeutung für das Unternehmen haben können. Die übertragung in ähnliche Problemfelder in anderen Betrieben scheint zumindest einer näheren Analyse wert. So ist vorstellbar, daß z.B. in Unternehmen des Maschinen- bzw. Anlagenbaus Probleme auftreten, die denen der Konfigurierung von Computersystemen strukturell ähneln. Die Systeme IMACS, ISA und ILOG lösen Probleme, wie sie in vielen Industriebetrieben in ähnlicher Weise vorkommen.

Das Beispiel zeigt aber auch deutlich, wie fruchtbar die Zusammenarbeit zwischen Hochschulen und Wirtschaftsunternehmen für beide Seiten sein kann. Man kann davon ausgehen, daß DEC das Problem der Konfigurierung mit Hilfe eines Expertensystems zumindest zum damaligen Zeitpunkt ohne Hilfe der Carnegie Mellon Universität nicht hätte bewältigen können. Andererseits hat gerade die Entwicklung von XCON der Forschung erhebliche Impulse vermittelt - dies gilt in besonderem Maße für die Erforschung von Anwendungsmöglichkeiten im Bereich der Betriebswirtschaft.

3.4.2 Die Integration in die Anwendungsumgebung

Sollen Expertensysteme nicht nur für Experimente oder theoretische Studien entwickelt, sondern auf konkrete praktische Probleme angewandt werden, so müssen sie in die Anwendungsumgebung integriert werden[1]. Diese Forderung wirft einige technische und organisatorische Fragestellungen auf, die in diesem Abschnitt analysiert werden sollen.

Betriebswirtschaftliche Probleme zeichnen sich oft durch eine große Menge zu berücksichtigender Daten aus, die teilweise im Rahmen der normalen betrieblichen Datenerfassung, z.B. bei der Buchhaltung, bereits in maschinell lesbarer Form anfallen. Die Menge dieser Daten schließt eine interaktive Eingabe durch den Benutzer während der Konsultation in der Regel aus. Aus diesem Grund ist zu fordern, daß Expertensysteme auf die betrieblichen Datenbestände unmittelbar zugreifen können[2].

Diese Forderung stellt Anforderungen an die Hard- und Softwareumgebung des Systems. Auf der Hardware-Seite muß eine Schnittstelle zwischen dem Rechner, auf dem das Expertensystem implementiert ist, und der betrieblichen EDV-Anlage hergestellt werden, sofern diese nicht identisch sind. Auf der Software-Seite muß einerseits das zur Konstruktion des Expertensystems verwendete Hilfsmittel[3] einen Zugriff auf externe Datenbestände ermöglichen. Dies kann z.B. durch Einbinden von in einer allgemeinen Programmiersprache entwickelten Unterprogrammen in das Expertensystem geschehen[4].

---

1) Vgl. z.B. Roesner, Savory (73), S.28 f.
2) Vgl. Schreier (78), S.30.
3) Vgl. Abschnitt 2.4.
4) Vgl. Abschnitt 2.5.7.

Andererseits muß die Software der betrieblichen EDV-Anlage so beschaffen sein, daß ein Zugriff auf die Datenbestände über eine Schnittstelle von außen möglich ist.

An dieser Stelle soll nicht näher auf die technischen Details dieser Problematik eingegangen werden[1]. Vielmehr sollen im folgenden einige typische Hard-/Softwarekonstellationen beschrieben und die damit verbundenen Integrationsprobleme aufgezeigt werden. Es lassen sich dabei drei mögliche Konstellationen unterscheiden, die im konkreten Fall noch variieren können:

1) Das Expertensystem wird in einer speziell dafür geschaffenen Hard-/Softwareumgebung, z.B. mit Hilfe einer Shell auf einer sog. LISP-Maschine[2], erstellt. Hard- und Software bieten nicht die Möglichkeit, Schnittstellen zur betrieblichen EDV-Anlage herzustellen.

Dieser Fall tritt heute noch relativ häufig auf. Allerdings haben die Hersteller einschlägiger Hard- und Software das Schnittstellenproblem erkannt und bieten in der Regel zu den gängigen EDV-Anlagen Schnittstellen an. Probleme der gezeigten Art treten also in erster Linie dann auf, wenn die betriebliche EDV-Anlage älteren Datums ist oder von einem kleineren Hersteller stammt.

Eine Integration in das betriebliche Informationssystem kann in diesem Fall nur erfolgen, wenn das Expertensystem auf der betrieblichen EDV-Anlage neu implementiert

---

1) Vgl. dazu z.B. Schreier (78).
2) Hierunter versteht man einen speziell auf die Symbolverarbeitung abgestimmten Rechenautomaten, dessen Betriebssystem in der Programmiersprache LISP (Vgl. Abschnitt 2.4.3) implementiert ist. Derartige Rechner verfügen in der Regel über eine eingebaute, sehr leistungsfähige Benutzerschnittstelle und weisen eine besonders hohe Rechengeschwindigkeit bei Anwendungen der künstlichen Intelligenz auf.

wird. Dies bedeutet, daß das System zunächst in der komfortablen Entwicklungsumgebung erstellt und verbessert wird. Anschließend wird es dann in einer konventionellen Programmiersprache auf der betrieblichen EDV-Anlage neu programmiert, wobei in der Regel auf aufwendige Hilfskomponenten wie eine natürlichsprachliche Benutzerschnittstelle oder die Wissensakquisitionskomponente verzichtet wird[1].

Diese Methode hat den Vorteil, daß ein Expertensystem in einer komfortablen Umgebung mit relativ geringem Aufwand erstellt werden kann. Das übertragene System kann dann die speziellen Eigenschaften der betrieblichen EDV-Anlage optimal nutzen und so z.B. zu relativ kurzen Antwortzeiten führen. Der Nachteil liegt zum einen in dem erheblichen Aufwand, den die Neuimplementierung darstellt, zum anderen in der schwierigen Aktualisierbarkeit des resultierenden Expertensystems, womit gerade einer der wichtigsten Vorteile dieser Technologie aufgegeben wird.

2) Das Expertensystem wird in spezieller Hard- und Software implementiert, es besteht jedoch die Möglichkeit, Schnittstellen zur betrieblichen EDV-Anlage herzustellen.

In dieser Situation ist eine übertragung des fertigen Expertensystems auf die betriebliche EDV-Anlage dennoch möglich. Dies hat den Vorteil, daß die vorhandene Hardware genutzt werden kann und nicht für jeden Expertensystem-Arbeitsplatz ein eigener Spezialrechner angeschafft werden muß, was heute noch eine erhebliche Investition darstellen würde[2]. Man handelt sich aller

1) Vgl. Abschnitt 2.4.3.
2) Ein derartiger Rechner kostete, wie bei einer Präsentation eines namhaften Herstellers zu erfahren war, 1986 inklusive entsprechender Software 50 bis 150 TDM.

dings die oben beschriebenen Nachteile ein und verzichtet zudem auf die sehr komfortable Anwendungsumgebung, die Teil derartiger Spezialrechner ist und die die Akzeptanz des Expertensystems möglicherweise erhöhen könnte.

3) Das Expertensystem soll direkt auf der betrieblichen Hardware erstellt werden, wobei gegebenenfalls geeignete Unterstützungssoftware beschafft werden kann.

Dieser Fall ist aus der Sicht der Integration in das betriebliche Informationssystem am wenigsten problematisch. Ob diese Vorgehensweise sinnvoll ist, hängt allerdings entscheidend davon ab, welche Unterstützungssoftware für die jeweilige Rechenanlage erhältlich ist. Zur Zeit sind vor allem für konventionelle Personal Computer zahlreiche Hilfsmittel zur Erstellung von Expertensystemen erhältlich. Allerdings bieten diese Rechner oft nicht die für eine größere praktische Anwendung notwendige Leistungsfähigkeit.

Für konventionelle EDV-Anlagen ist z.B. am Lehrstuhl für Betriebsinformatik der Universität Erlangen-Nürnberg die Expertensystem-Shell HEXE entwickelt worden[1]. Das System wurde in der auf betrieblichen EDV-Anlagen sehr weit verbreiteten Sprache COBOL implementiert, die zwar eine Konstruktion von Expertensystemen nicht erleichtert, aber eine Integration in das betriebliche Informationssystem unterstützt. Das System wurde zur Konstruktion einer Vielzahl von Expertensystemen eingesetzt, von denen einige sich inzwischen im praktischen Einsatz befinden[2].

1) Vgl. Mertens, Allgeyer, Das (53), S. 922 f.
2) Vgl. ebenda, S. 923 ff.

Tendenziell läßt sich feststellen, daß eine Entwicklung von Expertensystemen auf konventionellen Rechenanlagen wesentlich schwieriger ist als auf Spezialrechnern. Dies liegt daran, daß letztere eine speziell auf die Symbolverarbeitung zugeschnittene Architektur aufweisen. Eine komfortable Benutzeroberfläche unter Zuhilfenahme von Fenstertechnik, hochauflösender graphischer Darstellung und großflächigen Bildschirmen erleichtert die Wissensakquisition und das Suchen von Fehlern. Andererseits sind konventionelle Rechenautomaten mit ihrer auf die Zahlenverarbeitung zugeschnittenen Architektur besser geeignet, um betriebliche Datenbestände quantitativ zu analysieren, so daß der Vorteil der Spezialrechner bei betriebswirtschaftlichen Anwendungen zum Teil aufgewogen wird.

Neben der aufgezeigten Problematik des Zugriffs auf betriebsinterne Datenbestände ergibt sich das Problem einer Integration des Systems in bestehende organisatorische Abläufe[1]. Diese vielschichtige Problematik kann hier nur kurz umrissen werden.

Zunächst müssen die Benutzer im Umgang mit dem Expertensystem geschult werden. Sie müssen lernen, es als Hilfsmittel in den täglichen Arbeitsablauf einzugliedern wie etwa ein gewöhnliches Nachschlagewerk. Dabei muß vor allem darauf geachtet werden, daß sich die Benutzer der Grenzen des Systems bewußt sind und die Ergebnisse nicht unreflektiert übernehmen[2].

Weiterhin muß die Zugangsberechtigung zum System geregelt werden. Insbesondere ist festzulegen, wer Zugriff auf die Wissensakquisitionskomponente erhalten soll,

---

1) Vgl. Roesner (72), S.43 f.
2) Vgl. Abschnitt 3.2.3.

d.h. wer die Wissensbasis ändern darf, sowie ein geeignetes Dokumentationssystem zu entwerfen, so daß Änderungen jederzeit nachvollziehbar sind. Die Verantwortung für die Aktualisierung der Wissensbasis muß an geeignete Mitarbeiter übertragen werden.

Schließlich kann die Einführung eines Expertensystems gravierende Änderungen der betrieblichen Organisationsstruktur bewirken. Kompetenzen und Verantwortung können sich verschieben, wobei ein Expertensystem tendenziell eine Dezentralisierung von Entscheidungskompetenzen, unter den in Abschnitt 3.2.2 beschriebenen Einschränkungen, begünstigt.

Im Vergleich zur Einführung eines konventionellen EDV-Systems können jedoch die organisatorischen Änderungen vergleichsweise gering gehalten werden. Dies liegt zum einen an der extremen Flexibilität des Ansatzes der Wissensverarbeitung, der kein bestimmtes Format für die zu verarbeitenden betrieblichen Daten festlegt. Es müssen daher in der Regel keine Änderungen in der betrieblichen Datenerfassung bzw. -aufbereitung vorgenommen werden.

Zum anderen ermöglicht die schrittweise Vorgehensweise des Knowledge Engineering ein langsames "Hereinwachsen" des Expertensystems in die vorhandenen Strukturen. Es muß nicht "von heute auf morgen" ein neues System eingeführt werden. Vielmehr kann das Expertensystem zunächst z.B. an einem einzelnen Arbeitsplatz auf einen eingeschränkten Problembereich angewandt werden. Bewährt es sich dort, können weitere Arbeitsplätze mit dem System ausgestattet werden und der Problembereich bzw. die Wissensbasis kann erweitert werden. Auf diese Weise können die Benutzer behutsam an die neue Technologie herangeführt werden.

3.4.3 Expertensysteme als Beitrag zu "Intelligenten Management-Informations-Systemen"

Nach Koreimann versteht man unter einem Management-Informations-System "die organisatorische Konzeption des gesamten betrieblichen Informationswesens in dem Sinne, daß das Management die für die Durchführung seiner Aufgaben benötigten Informationen über die Vergangenheit, über das Ist und über die Zukunft (Prognosen) entsprechend dem Zweck (Situation) mit dem richtigen Inhalt, zum richtigen Zeitpunkt in der zweckmäßigsten Form unter Berücksichtigung des allgemeinen Wirtschaftlichkeitsprinzips zur Verfügung hat."[1]

Ziel dieser Systeme ist es demnach, dem Management Informationen und Entscheidungshilfen zu liefern[2]. Die konventionellen Management-Informations-Systeme stellen dazu eine Reihe von Methoden zur Verfügung, die das im Betrieb sowie in Marktforschungs-Studien anfallende Primärmaterial zu entscheidungsrelevanten Größen verdichten sollen. Das Spektrum dieser Methoden reicht von statistischen Analysen wie z.B. der Regressionsanalyse bis hin zu den Verfahren der Unternehmensforschung wie etwa der Simulation.

Die eingesetzten Methoden enthalten implizit eine Reihe von Voraussetzungen, die der Benutzer erfüllen muß, um das Management-Informations-System sinvoll zu nutzen[3]:

> Der Benutzer muß wissen, was für ein Problem er eigentlich hat, d.h. er muß die formale Struktur des seiner konkreten Fragestellung zugrundeliegenden Problems kennen, um die adäquate Methode auswählen zu können.

---

1) Koreimann (42), S.21.
2) Vgl. ebenda.
3) Vgl. Cooper (15), S.219 f.

Der Benutzer muß in der Lage sein, das reale Problem in einem formalen Modell abzubilden. Dies bedeutet in der Regel, daß er die Beziehungen zwischen den verschiedenen Einfluß- bzw. Entscheidungsgrößen in Form von mathematischen Gleichungen angeben kann.

Der Benutzer muß rechtzeitig Zugang zu allen relevanten Informationen haben.

Der Benutzer muß in der Lage sein, die Ergebnisse des Modells vor dem Hintergrund der realen Situation richtig zu interpretieren. Dies ist in der Regel nur dann möglich, wenn er die angewandte Methode mit den darin enthaltenen impliziten Annahmen und Voraussetzungen vollständig verstanden hat.

Der Benutzer muß den Umgang mit den in der Regel in Form von Computerprogrammen vorliegenden Methoden beherrschen. Dies setzt oft eine längere Einarbeitung in das entsprechende Programm sowie grundlegende Kenntnisse im Umgang mit Computern voraus.

Demgegenüber zeichnen sich die Adressaten der Management-Informations-Systeme, nämlich die Manager, meistens dadurch aus[1],

daß sie ihre Entscheidungsprobleme nur empirisch verbal, d.h. nicht formal, beschreiben können;

daß sie nicht über die Zeit oder Kenntnisse verfügen, um mathematische Modelle konstruieren und auswerten zu können;

daß sie nicht rechtzeitig Zugang zu allen relevanten Informationen haben;

1) Vgl. Cooper (15), S.217.

- daß sie die Methoden der Management-Informations-Systeme nicht richtig verstehen und deshalb Schwierigkeiten haben, die Ergebnisse richtig zu interpretieren;

  daß sie im Umgang mit den entsprechenden Programmen sowie mit Computern im allgemeinen nicht oder zu wenig vertraut sind, um die Methoden anwenden zu können;

  sowie daß sie allgemein zu wenig Zeit haben, die Methoden der Management-Informations-Systeme anzuwenden, selbst wenn sie ansonsten alle Voraussetzungen dazu mitbrächten.

Die mangelnde übereinstimmung zwischen den praktischen Erfordernissen eines Management-Informations-Systems und den Leistungen der tatsächlich vorhandenen Systeme hat dazu geführt, daß diese Systeme nicht in ausreichendem Maß genutzt werden[1]. Es ist nun zu fragen, in wieweit Expertensysteme dazu beitragen können, die auf gezeigte Kluft zwischen den Managern und den modernen Methoden der Entscheidungsunterstüzung zu überbrücken.

Im folgenden soll der Entwurf eines "Intelligenten Management-Informations-Systems" skizziert werden. Damit ist ein Informationssystem gemeint, das eine gesuchte Information dadurch beschafft, daß es Wissen über die für diese Informationen relevanten Daten und Auswertungsmethoden anwendet[2]. Die Bezeichnung "intelligent" soll keinen Hinweis auf eine bestimmte Leistungsfähigkeit geben, sondern eine Abgrenzung zu den her

1) Vgl. Cooper (15), S.217.
2) Auf die Möglichkeiten des Einsatzes eines solchen Systems für betriebswirtschaftliche Fragestellungen weist Mertens in (51), S.27, hin.

kömmlichen Systemen bilden und auf die Tatsache hinweisen, daß Methoden der künstlichen Intelligenz zu seiner Realisierung verwendet werden.

In der Informationswissenschaft werden intelligente Informationssysteme als Ansatz zur Erweiterung konventioneller Information-Retreival-Systeme gesehen, wobei "intelligent" sich vor allem auf die Fähigkeit bezieht, Benutzeranfragen in natürlicher Sprache zu interpretieren und die Ergebnisse in geeigneter Weise darzustellen[1]. Demgegenüber enthält ein intelligentes Management-Informations-System der im folgenden vorgestellten Art neben der Fähigkeit, auf betriebliche Datenbestände zuzugreifen, auch Komponenten zur numerischen Auswertung dieser Daten, z.B. in Form von Regressionsanalysen.

Abb. 3.5 stellt den Aufbau eines derartigen Systems dar. Eine Benutzerschnittstelle analysiert die natürlichsprachlichen Anfragen des Benutzers und leitet daraus die relevanten Fragestellungen ab. Auf der Basis dieser Fragestellung ermittelt ein Expertensystem die zur Beantwortung der Frage notwendigen Daten und Methoden und gibt eine entsprechende Anforderung an eine Auswertungskomponente weiter. Diese wendet die vom Expertensystem ermittelten, in einer Methodenbank gespeicherten Methoden auf die in der betrieblichen Datenbank vorhandenen Daten an und ermittelt das Ergebnis zurück an das Expertensystem.

Vor dem Hintergrund der ursprünglichen Fragestellung liefert das Expertensystem eine Interpretation der Ergebnisse, die von der Benutzerschnittstelle in einen

1) Vgl. z.B. Hahn (31), Appelrath (2).

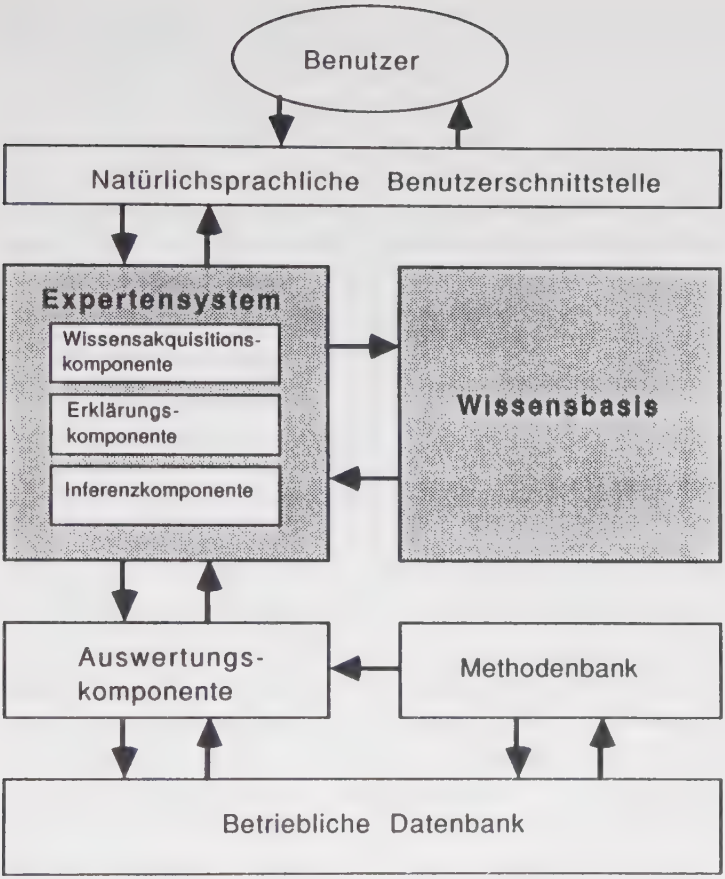


Abb. 3.5: Aufbau eines "Intelligenten Management-Informations-Systems"

natürlichsprachlichen Text umgewandelt wird. Dieser Text wird an den Benutzer ausgegeben. Auf Wunsch kann das Zustandekommen des Ergebnisses mit Hilfe der Erklärungskomponente des Expertensystems nachvollzogen werden.

Zur Veranschaulichung soll die Arbeitsweise eines intelligenten Management-Informations-Systems an einem einfachen, fiktiven Fallbeispiel dargestellt werden:

Der Manager eines Unternehmens der Textilbranche könnte an ein intelligentes Informationssystem etwa die folgende Frage richten: "Warum ist der gesamte Umsatz in diesem Monat im Vergleich zum Vormonat zurückgegangen?". Das System würde zunächst den natürlichsprachlichen Satz in eine interne Form übertragen, die etwa folgenden Inhalt haben könnte: Erkläre_Differenz (Gesamtumsatz (März), Gesamtumsatz (April)).

Das System würde nun in seiner Wissensbasis nach einer Erklärung des Begriffs "Gesamtumsatz" suchen und dabei vielleicht darauf stoßen, daß sich der Gesamtumsatz aus den Umsätzen von vier Produktarten zusammensetzt. Diese mögen im Beispiel eines Textil-Herstellers die Produkte "Wintermäntel", "Sommermäntel", "Unterwäsche" und "Haustextilien" sein. Das System würde nun versuchen, die Entwicklung des Gesamtumsatzes durch die Entwicklung der Einzelumsätze zu erklären.

Beispielsweise könnte die Auswertungskomponente aus den vorhandenen betrieblichen Daten ermitteln, daß der Umsatz von Wintermänteln seit dem Vormonat stark zurückgegangen ist, während Sommermäntel auf einem gleichbleibend niedrigen, Unterwäsche auf einem durchschnittlichen und Haustextilien auf einem leicht gestiegenen Umsatzniveau waren. Eine weitere Suche in der Wissensbasis könnte erbringen, daß für die Jahreszeit ein starker Rückgang der Wintermäntel, ein mäßiger Anstieg

bei Sommermänteln und ein gleichbleibender Verkauf der übrigen Produkte zu erwarten ist.

Durch einen Vergleich stellt das System fest, daß sich lediglich der Umsatz der Sommermäntel für die Jahreszeit stark untypisch verhält. Es konzentriert nun seine Ursachensuche auf diesen Bereich, indem es die Gründe für den ausgebliebenen Umsatzanstieg herauszufinden versucht. Dazu zieht es weiteres Wissen aus der Wissensbasis heran, das etwa folgenden Inhalt haben könnte:

"Wenn sich der Umsatz eines Produktes untypisch verhält, wende eine Regressionsanalyse auf die saisonbereinigten Verkaufszahlen des Produktes der letzten zwölf Monate an. Liegt ein kontinuierlicher Rückgang vor, dann prüfe die Hypothesen "schlechter Außendienst", "zu wenig Werbung" und "mangelnde modische Aktualität". Anderenfalls prüfe die Hypothese "schlechtes Wetter"."

Das Expertensystem würde nun die Auswertungskomponente aktivieren, die die Methode "Regressionsanalyse" aus der Methodenbank lädt und auf die Umsatzwerte für Sommermäntel der letzten zwölf Monate anwendet, wobei evtl. zuvor die Methode "Saisonbereinigung" auf diese Daten anzuwenden wäre. Die resultierenden Regressionskoeffizienten würden an das Expertensystem zurückübermittelt.

Schließlich wird das System einen Erklärungstext etwa der folgenden Form ausgeben:

"Der Rückgang des Gesamtumsatzes von März auf April erklärt sich aus einem starken Rückgang des Absatzes an Wintermänteln bei gleichbleibendem Geschäft im Bereich der Unterwäsche und Sommermäntel und trotz leicht steigenden Absatzes der Haustextilien. Dieses Ergebnis ist insgesamt negativ zu beurteilen, da für den Bereich der

Sommermäntel ein Absatzanstieg zu erwarten war. Der schlechte Absatz der Sommermäntel erklärt sich zum Teil durch das anhaltend schlechte Wetter."

Die Ausgabe eines natürlichsprachlichen Erklärungstextes wird durch die Verwendung von Textbausteinen ermöglicht. Dabei werden je nach Ausprägung einzelner Variablen verschiedene Textteile zu einem Gesamttext zusammengefügt. Je nach Sachverhalt könnten so an einer bestimmten Textstelle alternativ die Worte "steigend", "gleichbleibend", "stark fallend" usw. stehen. Beispiele für die Verwendung von Textbausteinen zur Erzeugung natürlichsprachlichen Outputs sind die Expertensysteme BILEX und GUVEX zur Bilanz- bzw. GuV-Analyse, die an der Universität Erlangen-Nürnberg unter der Leitung von Prof.Dr. P.Mertens entwickelt werden[1].

Das Beispiel demonstriert einige Vorteile, die ein derartiges System gegenüber konventionellen Management-Informations-Systemen hätte:

Der Benutzer kann seine Anfrage in natürlicher Form stellen, ohne Wissen über die zugrundeliegende Problemstruktur haben zu müssen.

Der Benutzer braucht kein kompliziertes mathematisches Modell zu erstellen.

Der Benutzer braucht kein Wissen über die angewendeten Methoden, wie etwa die Regressionsanalyse, zu haben, um die Ergebnisse interpretieren zu können.

Durch die natürlichsprachliche Benutzerschnittstelle braucht der Benutzer keine Erfahrung im Umgang mit Computern oder spezieller Software zu haben, um das System nutzbringend einsetzen zu können.

1) Vgl. Dräger u.a. (19), Büttner u.a. (12).

- Das System kann zwischen "auffälligen" (der starke Rückgang der Wintermäntel) und "wichtigen" (der gleichbleibende Absatz der Sommermäntel) Informationen unterscheiden und so die Ursachensuche auf die relevanten Bereiche konzentrieren. Auf diese Weise wird sowohl ein "Zahlenberg" als auch ein zu hoher Informationsverlust vermieden.

- Mit Hilfe der Erklärungskomponente des Expertensystems kann der Fortgang der Analyse jederzeit verfolgt und die Auswahl von bestimmten Methoden begründet werden.

- Der unkomplizierte Umgang mit dem System führt zu einem erheblichen Zeitgewinn bei der Entscheidungsfindung und somit zu größerer Flexibilität[1].

Trotz der dargestellten Vorteile sind einige Einschränkungen in Bezug auf die Anwendbarkeit eines derartigen Systems zu machen. Zunächst erscheint es zum gegenwärtigen Zeitpunkt nicht möglich, ein System zu erstellen, das die gesamte Spannweite betriebswirtschaftlicher Fragestellungen umfaßt. Ebenso müßte sich ein derartiges System zunächst auf eine kleine Auswahl von Methoden beschränken, um die Problemkomplexität in einem handhabbaren Rahmen zu halten.

Schließlich kann ein solches System sinnvoll nur dann eingesetzt werden, wenn die betriebsindividuellen Gegebenheiten in der Wissensbasis berücksichtigt sind. Es ist jedoch fraglich, ob der hohe Aufwand für die Erstellung eines Expertensystems nur für einen einzelnen Betrieb gerechtfertigt wäre. Aus diesem Grund wären intelligente Management-Informations-Systeme vermutlich nur dann wirtschaftlich sinnvoll realisierbar, wenn es

1) Vgl. von Zelewski (99), S.731 ff.

gelänge, eine "vorstrukturierte Wissensbasis" zu schaffen, in der das Grundwissen über die verwendeten Methoden und die allgemeinen betriebswirtschaftlichen Zusammenhänge enthalten wäre. Diese Wissensbasis könnte dann mit vertretbarem Aufwand an die jeweilige betriebliche Situation angepaßt werden.

Zusammenfassend läßt sich sagen, daß ein umfassendes intelligentes Management-Informations-System in der dargestellten Form noch nicht realisierbar ist. Die hier dargestellten Grundgedanken sind jedoch teilweise bereits in Expertensystemen realisiert worden. So wurde beispielsweise am Institut für Informatik der Technischen Universität München in Zusammenarbeit mit der Hypo-Bank ein Expertensystem zur Filialanalyse entwickelt[1].

Aufgabe dieses Systems ist die Beurteilung von Kennzahlen der Filialen der Bank, wobei es in erster Linie darum geht, "Signifikanzfälle" zu ermitteln, d.h. solche Fälle, bei denen auffällige Differenzen zwischen einer "Vaterkennzahl" und einer wichtigen "Sohnkennzahl" auftreten. Der Bewertungsexperte kann seine Aufmerksamkeit auf solche "Ungereimtheiten" konzentrieren und wird so in seiner Arbeit wesentlich entlastet.

Ein weiteres Beispiel ist der von der London School of Economics durchgeführte Versuch, den Benutzer mit Hilfe von Expertensystemen und natürlichsprachlicher Benutzerschnittstelle bei der Formulierung von ereignisorientierten Simulationsmodellen zu unterstützen[2]. Die dabei entwickelten Prototypen zeigen, daß Expertensysteme die Entwicklung von Computer-Modellen wesentlich erleichtern können.

---

1) Vgl. Doukidis, Paul (18).
2) Vgl. Güntzer u.a. (30).

Die angeführten Beispiele bestätigen, daß Expertensysteme eine entscheidende Hilfe dabei sein können, die modernen Methoden der Entscheidungsunterstützung dem Management einfacher und effektiver zugänglich zu machen. Dies könnte einen wesentlichen Einfluß auf die Verbesserung der Informationsgrundlage unternehmerischer Entscheidungen in der Praxis haben.

4 FOUNDER: Ein Expertensystem zur Existenzgründungs-Beratung

4.1 Existenzgründung als Entscheidungsproblem

4.1.1 Problemstellung

Der Aufbau einer selbständigen Existenz als Unternehmer stellt für den Gründer einen bedeutenden, folgenschweren und risikoreichen Schritt dar. Im Unterschied zur Gründung eines Unternehmens durch eine bereits vorhandene Wirtschaftseinheit (unselbständige Unternehmensgründung[1]) spielen bei der Existenzgründung (selbständige Unternehmensgründung[2]) neben betriebswirtschaftlichen und formaljuristischen auch personenbezogene und psychologische Faktoren eine Rolle[3].

Unter dem Begriff "Existenzgründung" soll in diesem Zusammenhang die Gründung eines eigenständigen Gewerbebetriebes durch eine oder mehrere Gründerpersonen zum Zwecke der Sicherung des Lebensunterhaltes der Gründer verstanden werden. Die Gründung freiberuflicher Praxen soll in diesem Sinne ebenfalls als Existenzgründung verstanden werden, auch wenn diese nicht unter den Begriff des "Gewerbebetriebes" nach dem Gewerberecht fallen[4].

Der weitaus größte Teil der Existenzgründungen in diesem Sinne entfällt auf die Gründung von Klein- und Kleinstbetrieben wie Einzelhandelsgeschäften, Handwerks- und Kleingewerbe-Betrieben. Diese Gründungen sind von der betriebswirtschaftlichen Literatur bisher

1) Vgl. Szyperski, Nathusius (88), S.27 ff.
2) Vgl. ebenda.
3) Vgl. ebenda, S.87.
4) Vgl. §6 GewO.

kaum explizit untersucht worden, weil davon ausgegangen wird, daß hier dieselben Problemstellungen anzutreffen seien wie bei bereits existierenden Unternehmen[1]. Dies kann jedoch schon deshalb angezweifelt werden, weil die Gründer in der Regel nur über geringe betriebswirtschaftliche Kenntnisse und keinerlei Erfahrung in der Betriebsführung verfügen. Ihnen steht damit das umfangreiche Instrumentarium der Betriebswirtschaftlehre zur Lösung von Entscheidungsproblemen in der Regel nicht zur Verfügung.

Zudem kommt auf den Gründer eine Vielzahl von nahezu gleichzeitig zu treffenden Entscheidungen zu, die oft auf Jahre hinaus wirken und somit eindeutig strategischen Charakter haben[2]. Dazu gehört die Wahl der Branche, des Produkt- bzw. Leistungsprogramms, des Standortes und der Rechtsform, um nur einige beispielhaft herauszugreifen. Diese Entscheidungen sind unter erheblichem Zeitdruck und in der Regel auf der Basis einer äußerst mangelhaften Informationsgrundlage zu treffen.

In den folgenden Abschnitten soll zunächst dargestellt werden, welche Entscheidungsbereiche bei der Existenzgründung eine Rolle spielen und welche praktischen Probleme dabei auftreten. Dann soll ein Expertensystem Ansatz entwickelt werden, mit Hilfe dessen die Informationsgrundlage des Existenzgründers verbessert werden kann.

---

1) Vgl. Szyperski (87), S.152.
2) Vgl. Szyperski (87), S.153 f., Rentrop (69), S.11 f

4.1.2 Der Prozeß der Existenzgründung

Abb. 4.1 zeigt den Prozeß der Existenzgründung von der Gründungsidee bis zur Geschäftsaufnahme[1]. Er kann als mehrstufiger Planungsprozeß aufgefaßt werden, wobei zwischen die einzelnen Planungsstufen "Filter" in Form von Überprüfungen durch den Gründer oder Dritte geschaltet sind.

Es muß jedoch darauf hingewiesen werden, daß dieses eine idealisierte Darstellung ist. In der Praxis werden die einzelnen Planungsstufen in der Regel nicht in klarer Abgrenzung oder festgelegter Reihenfolge durchgeführt. Ebenso ist der Detailliertheitsgrad der Planung nicht einheitlich: während einige Details minutiös geplant werden, werden andere Entscheidungen oberflächlich oder gar zufällig getroffen.

Am Beginn des Entscheidungsprozesses steht eine Gründungsidee. Diese umfaßt meist eine grobe Vorstellung der Branche und des angestrebten Produkt- oder Leistungsprogramms. Teilweise (z.B. bei Erfindungen) stehen auch bereits sehr konkrete Vorstellungen über Produkte oder Dienstleistungen am Anfang der Überlegungen.

Diese Idee muß in einem ersten Schritt konkretisiert werden, indem der notwendige Aufwand für die Realisierung sowie der zu erwartende Ertrag grob geschätzt werden. Hierzu werden Informationen darüber benötigt, welche Mittel (in Form von Kapital, Geschäftsausstattung, Maschinen, Räumen, Personal etc.) zur Realisierung der Idee erforderlich sind. Aus diesen Angaben lassen sich der Kapitalbedarf sowie die laufenden Betriebskosten als Anhaltswerte ermitteln.

---

1) Vgl. Szyperski, Nathusius (88), S.33. Die folgende Darstellung des Gründungsprozesses beruht zum Teil auf empirischen Beobachtungen, die im Verlauf der Experten-Interviews erarbeitet wurden; vgl. Abschnitt 4.2.3.

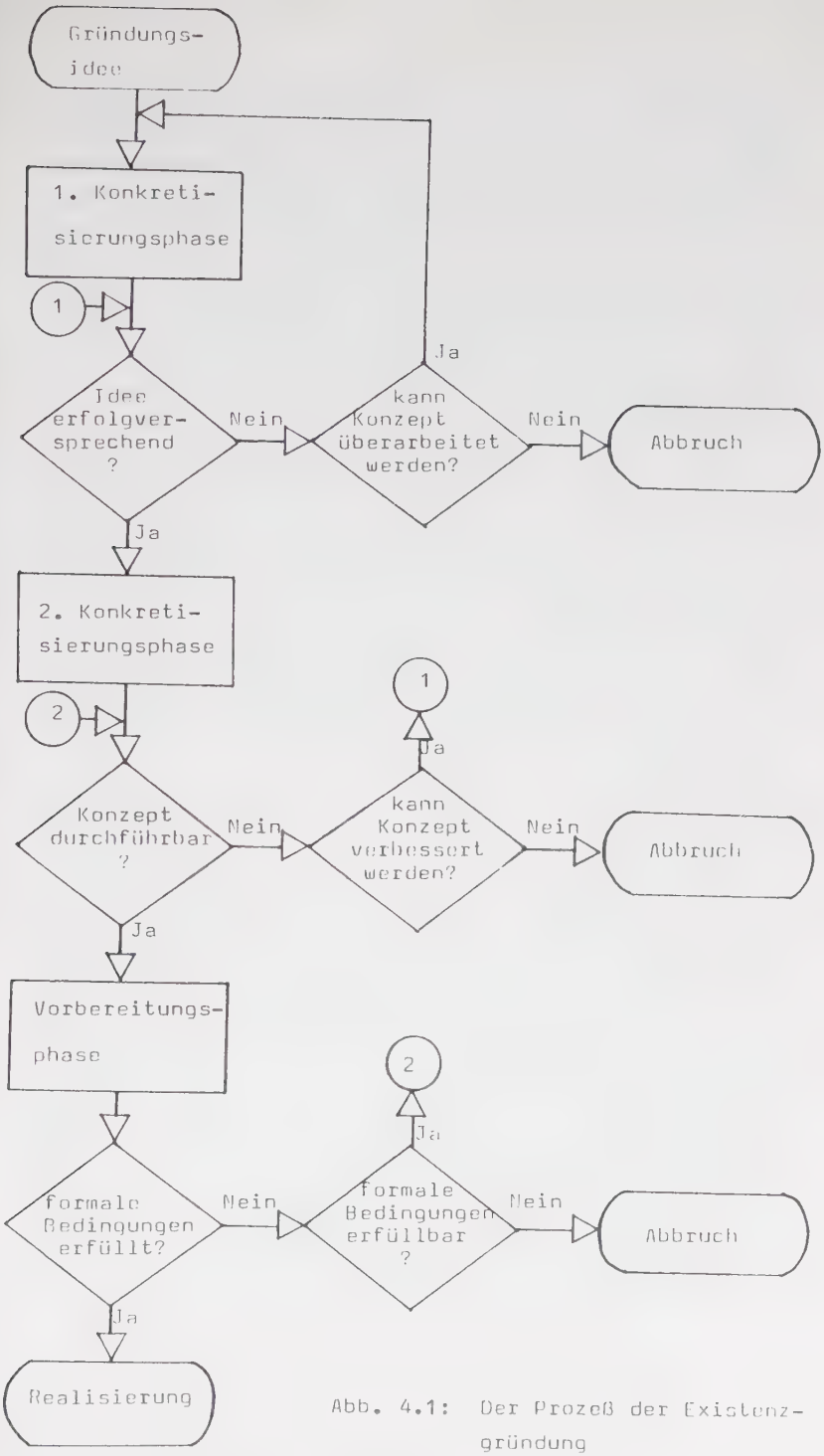


Abb. 4.1: Der Prozeß der Existenzgründung

Schließlich werden Informationen darüber benötigt, welcher Ertrag bei der Realisierung der Idee zu erwarten ist. Dazu können zunächst Umsatzstatistiken der entsprechenden Branche als Richtwert herangezogen werden. Falls solche Werte nicht verfügbar sind (z.B. bei technologieorientierten Gründungen), muß eine grobe Vorkalkulation erfolgen, bei der realisierbare Preise und Absatzmengen für die Produkte oder Dienstleistungen geschätzt werden müssen.

Bereits in dieser ersten Konkretisierungsphase zeigt sich, wie wichtig eine Kenntnis des angestrebten Marktes für den späteren Unternehmenserfolg ist[1]. Selbst eine nur grobe, halbwegs realistische Einschätzung der erforderlichen Mittel ist kaum möglich, wenn man nicht auf die Erfahrungswerte eines vergleichbaren Betriebes (etwa durch Statistiken oder frühere Berufserfahrung) zurückgreifen kann. Eine realistische Abschätzung der Absatzchancen ohne fundierte Marktkenntnis erscheint gänzlich unmöglich.

Meist stellt sich bei einem ersten Vergleich von erwartetem Aufwand und Ertrag bereits heraus, daß die bei der ersten Gründungsidee gehegten Vorstellungen von einem schnellen Gewinn unerfüllbar sind. Sollte dennoch eine realistische Chance für einen Erfolg des Unternehmens erkennbar sein, muß eine zweite Konkretisierungsphase erfolgen, in der die im ersten Schritt angestellten Überlegungen weiter verfeinert und präzisiert werden.

Im zweiten Konkretisierungsschritt müssen umfangreiche Informationen eingeholt werden. Während die erste Konkretisierungsphase meist durch den Gründer allein er-

1) Vgl. Szyperski, Nathusius (88), S.40 f.

folgt, wird er in der zweiten Phase mit Außenstehenden über seine Pläne sprechen. Dies können Freunde oder Familienangehörige, aber auch öffentliche und private Beratungsstellen wie die Industrie- und Handelskammer oder ein Steuerberater sein.

Manchmal stellt sich bereits bei solchen Gesprächen heraus, daß die Vorstellungen des Gründers in einigen Bereichen falsch oder unrealistisch sind. In diesem Fall muß die Planung insgesamt überarbeitet oder Details geändert werden. Das Ergebnis der zweiten Konkretisierungsphase ist schließlich ein Konzept, aus dem sowohl Fragen des Marktes (wie Standort, Absatzprogramm und Werbestrategie) als auch der Organisation (Rechtsform, Partner) und der Finanzierung (Kapitalbedarf, Gewinnerwartung, Liquiditätsplan) hervorgehen sollten.

Dieses Konzept muß auf seine Realisierbarkeit hin geprüft werden. Dies sollte idealerweise durch unabhängige, in Gründungsproblemen erfahrene Dritte wie Unternehmensberater, Berater der Kammern usw. erfolgen. Ein wichtiger Prüfstein kann auch die Verhandlung mit einer Bank über die Vergabe der erforderlichen Kredite sein. Banken neigen jedoch gerade bei der Gründung von Kleinbetrieben mitunter dazu, die Absicherung ihrer Kredite durch private Sicherheiten der Absicherung durch ein solides Unternehmenskonzept vorzuziehen, und schenken somit den betriebswirtschaftlichen Gegebenheiten der Gründung relativ wenig Beachtung[1].

Falls sich bei der Durchführbarkeitsprüfung noch gravierende Mängel ergeben, muß das Konzept unter Umstän

1) Dies ist eine empirische Beobachtung, die im Laufe der Experteninterviews zur Erstellung des Systems FOUNDER herausgearbeitet wurde; vgl. Abschnitt 4.2.3. Sie bestätigte sich auch in Gesprächen des Verfassers mit den Kreditabteilungen verschiedener Banken.

den bis hin zur Gründungsidee noch einmal überarbeitet werden oder das Vorhaben fallengelassen werden. Erscheint jedoch das Konzept durchführbar und frei von gravierenden Schwächen oder Risiken, kann mit der Vorbereitung der Gründung begonnen werden.

Bevor schließlich die Geschäftstätigkeit aufgenommen und damit der Gründungsprozeß abgeschlossen werden kann, ist noch eine Vielzahl von konkreten Vorbereitungsmaßnahmen zu treffen. Dazu gehören z.B. die Beantragung von öffentlichen Krediten und Zuwendungen, die Einrichtung der Geschäftsräume, die Erstausstattung des Warenlagers und die Anmeldung bzw. die Beantragung einer Genehmigung bei den zuständigen Stellen.

Auch in diesem Stadium kann sich die Gründung noch als undurchführbar erweisen, z.B. wenn die öffentlichen Stellen die Erteilung einer Konzession oder der Gewerbeerlaubnis verweigern. Sind jedoch alle rechtlichen und wirtschaftlichen Vorbereitungsschritte unternommen, kann die Aufnahme der Geschäftstätigkeit erfolgen.

Mit diesem Zeitpunkt beginnt die sog. Anlaufphase, in der die Umsätze noch nicht die zur Aufrechterhaltung des Geschäftes erforderliche Höhe erreichen. Die Ursachen dafür liegen meist in einem noch niedrigen Bekanntheitsgrad innerhalb des Marktes und dem Fehlen einer Stammkundschaft. Die Anlaufphase wird auch oft als "Durststrecke" bezeichnet, da die Einnahmen aus Umsatztätigkeit nicht ausreichen, um die laufenden Ausgaben und die private Lebenshaltung des Gründers abzudecken. Nun erweist sich, ob die Finanzierungsplanung solide genug ist, um Liquiditätsengpässe zu vermeiden.

Die Anlaufphase dauert in der Regel zwischen sechs und vierundzwanzig Monaten, sie kann jedoch kaum im voraus bestimmt werden und variiert stark von Branche zu Branche. Nach Ablauf dieser Zeit pendelt sich der Umsatz,

sofern keine gravierenden Veränderungen der Marktverhältnisse auftreten, auf seinem langfristigen Niveau ein. Erst jetzt stellt sich heraus, ob die bei der Planung erhobenen Erwartungen realistisch waren und das Unternehmen tatsächlich die Existenz des Gründers langfristig tragen kann.

In den folgenden Jahren wächst das Unternehmen immer mehr in den Markt hinein und kann sich so den sich ständig ändernden Marktbedingungen besser anpassen. Dieser Effekt beruht auf der zunehmenden Erfahrung des Gründers mit seinem Unternehmen und den täglich zu treffenden betriebswirtschaftlichen Entscheidungen.

Unmittelbar nach der Anlaufphase ist das "Jungunternehmen" jedoch besonders anfällig gegen Marktveränderungen und -krisen, da diese Erfahrung noch fehlt. Erschwerend kommt hinzu, daß nach der Anlaufphase die Konkurrenz verstärkt auf den "Newcomer" aufmerksam wird und unter Umständen mit Abwehrmaßnahmen reagiert, gegen die sich der unerfahrene und unterkapitalisierte Jungunternehmer nur schwer wehren kann. Er muß nun beweisen, daß seine Energie, Kreativität und Flexibilität ausreichen, um gegenüber den etablierten Unternehmen zu bestehen.

4.1.3 Strategische Entscheidungsbereiche der Existenzgründung

Abb. 4.2 zeigt die verschiedenen strategischen Entscheidungsbereiche, die der Existenzgründer im Rahmen seiner Gründungsplanung gestalten muß[1]. Sie lassen sich in die Bereiche "Markt", "Organisation" und "Finanzierung" unterteilen, wobei der Begriff "Organisation" hier auch ggf. die Planung des Produktionsprozesses beinhaltet.

1) Vgl. Szyperski (87), S.153 f., Rentrop (69), S.16.

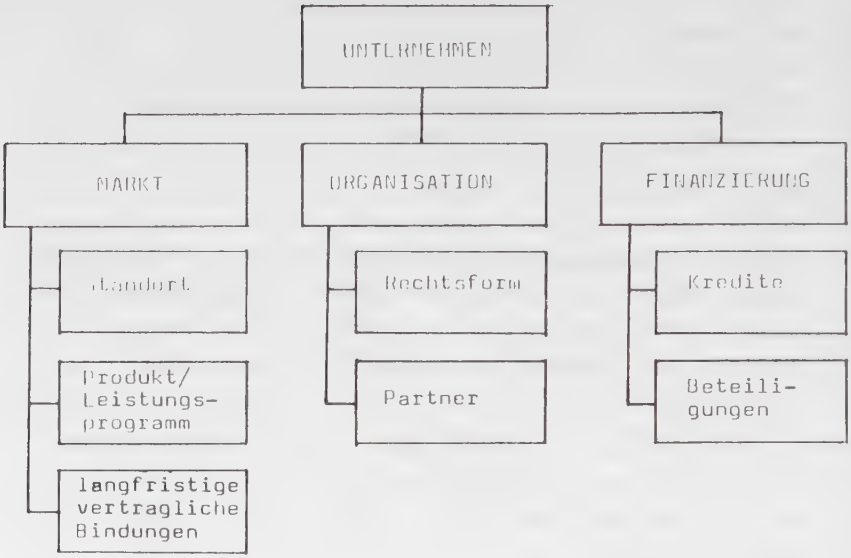


Abb. 4.2: Strategische Entscheidungsbereiche der Existenzgründung

Der strategische Gestaltungsspielraum des Gründers in bezug auf den Markt umfaßt die Wahl des Standortes, des Produkt- bzw. Dienstleistungsprogramms und, sofern langfristige vertragliche Bindungen eingegangen werden, bestimmter Lieferanten oder Abnehmer.

Von besonderer Bedeutung ist in den meisten Fällen die Wahl des Standortes. Hier ist eine Vielzahl von Faktoren zu beachten, die einen erheblichen Einfluß auf den Erfolg des neuzugründenden Unternehmens haben können[1]. Dazu zählen das Einzugsgebiet mit der dort vorhandenen Kundenstruktur, die Konkurrenzdichte am Standort, die

1) Zur Problematik der Standortwahl vgl. z.B. Nieschlag, Dichtl, Hörschgen (61), S.368 ff.

Verkehrsanbindung und sonstige Infrastruktur sowie die Mieten bzw. Pachten, um nur die wichtigsten zu nennen.

Das Gewicht, das den einzelnen Faktoren bei der Beurteilung eines potentiellen Standortes beizumessen ist, hängt entscheidend von der Art und Branche des Unternehmens ab. Dabei kommt naturgemäß der Standortwahl bei Einzelhandelsgeschäften eine größere Bedeutung zu als z.B. bei Produktionsbetrieben. Doch auch im letzteren Fall lohnt sich eine genaue Analyse verschiedener Standorte hinsichtlich Arbeitskräfteangebot, öffentlicher Förderungsmaßnahmen usw.

Trotz der elementaren Bedeutung der Standortwahl wird diese Frage von vielen Gründern zu wenig oder überhaupt nicht beachtet. An die Stelle rationalen Kalküls tritt häufig der Wunsch, sich in einer bestimmten, bereits bekannten Gegend anzusiedeln oder z.B. den bestehenden Freundeskreis zu erhalten[1]. Oft wird nicht einmal systematisch nach Standortalternativen gesucht, die dann miteinander verglichen werden könnten[2].

Die Wahl des Produkt- bzw. Dienstleistungsprogramms ist ebenfalls als strategische Entscheidung zu werten. Dabei sind Entscheidungen über die inhaltliche Umschreibung des Leistungsprogramms sowie über die Sortimentsbreite und Sortimentstiefe zu treffen[3]. Die inhaltliche Umschreibung des Leistungsprogramms legt die Branche sowie das Marktsegment fest. Sie ist nach Geschäftsaufnahme in der Regel gar nicht oder nur sehr schwierig zu ändern.

1) Vgl. Szyperski, Nathusius (88), S.60.
2) Vgl. ebenda.
3) Vgl. Meffert (49), S.371.

Entscheidungen über die Sortimentsbreite und Sortimentstiefe sind dagegen oft mittelfristig korrigierbar. Dabei ist jedoch zu beachten, daß der Markt auf Änderungen des Leistungsprogramms träge reagiert. Das bedeutet, daß alte Kunden, deren Erwartungen durch die Änderung enttäuscht wurden, verloren gehen, während neue Kunden nur mit zeitlicher Verzögerung gewonnen werden können, da die Änderung erst bei ihnen bekannt gemacht werden muß.

Hat ein Betrieb ein bestimmtes "Image" bei tatsächlichen und potentiellen Kunden (z.B. "breites Angebot" oder "Spezialanbieter mit großer Sortimentstiefe"), so bedarf es in der Regel eines erheblichen Aufwandes an Werbung und anderen Maßnahmen, um diesen Eindruck zu korrigieren. Aus diesem Grund ist es erforderlich, das Produkt- bzw. Dienstleistungsprogramm von vornherein auf ein bestimmtes Marktsegment auszurichten, um so in möglichst kurzer Zeit einen stabilen Kundenstamm aufzubauen. Auch diese Problematik wird jedoch von Gründern oft unterschätzt bzw. nicht erkannt.

Die Wahl des Produkt bzw. Dienstleistungsprogramms muß auch in Hinblick auf den Wettbewerb erfolgen. Gerade in diesem Bereich liegt die Chance, sich eine "Unique Selling Proposition"[1] (USP) zu schaffen, die es dem Gründer ermöglicht, sich von der Konkurrenz abzuheben. Ohne diese USP dürfte es schwerfallen, sich gegen die etablierte Konkurrenz zu behaupten.

Ein besonderes Problem gerade erfolgreicher Existenzgründungen ist es, daß das vom Gründer neu erschlossene Marktsegment plötzlich auch für größere, etablierte Firmen interessant wird, die sich bisher nicht in die-

1) Vgl. Szyperski, Nathusius (88), S.40.

sem Bereich betätigt haben. Diese können dann mit ihrer überlegenen Marktmacht den Gründer aus dem von ihm erschlossenen Marktsegment verdrängen, sofern er seinen Vorsprung als Innovator nicht halten kann[1]. Es ist deshalb ratsam, sich für die Existenzgründung ein Marktsegment auszuwählen, das entweder für größere Firmen nicht interessant genug ist oder in dem sich der Gründer aufgrund seiner höheren Flexibilität auch gegen übermächtige Konkurrenten behaupten kann[2].

Ebenfalls ein wichtiger strategischer Entscheidungsbereich ist die Frage vertraglicher Bindungen an Abnehmer oder Lieferanten, für die der sog. "Bierlieferungs-Vertrag" bei Gaststätten-Gründungen ein weit verbreitetes Beispiel ist. Derartige Verträge sind in der Regel mit erheblichen finanziellen Vorteilen wie günstigen Krediten verbunden. Sie legen jedoch den Gründer oft in wesentlichen Entscheidungsbereichen auf Jahre hinaus fest und nehmen ihm damit seine Flexibilität, die meist gerade eine der Stärken von Jungunternehmen ist. Auch diese Frage wird von vielen Gründern nicht fundiert analysiert. Häufig wird ein kurzfristiger Vorteil über einen langfristigen gestellt, was später unter Umständen mit der Insolvenz bezahlt werden muß.

Im Bereich der Unternehmensorganisation sind vor allem die Wahl der Rechtsform und die Aufnahme von Partnern als strategische Entscheidungen zu werten. Die Rechtsform des Unternehmens hat Auswirkungen auf die steuerliche Behandlung des Unternehmens sowie auf die Kreditwürdigkeit, wobei gilt, daß in der Regel eine Personengesellschaft (z.B. OHG) von den Banken besser beurteilt

1) Vgl. Szyperski, Nathusius (88), S.67 ff.
2) Vgl. ebenda.

wird als eine Kapitalgesellschaft (z.B. GmbH). Die Problematik der Eignung verschiedener Rechtsformen für Existenzgründungen soll hier jedoch nicht näher untersucht werden.

Die Aufnahme eines oder mehrerer Partner als persönlich haftende und damit entscheidungsberechtigte Teilhaber hat weitreichende Konsequenzen für die Existenzgründung. Zum einen birgt sie die Chance, Risiko und Belastungen der Gründung auf mehrere Personen zu verteilen und somit die persönlichen Belastungen des Gründers zu verringern. Außerdem können sich die verschiedenen Partner in ihren Stärken und Schwächen wie auch in finanzieller Hinsicht ergänzen und somit die wirtschaftliche Stärke und überlebensfähigkeit des Jungunternehmens deutlich erhöhen[1].

Andererseits birgt die Aufnahme von Partnern ein erhebliches Konfliktpotential. Das Risiko eines offenen Konflikts ist dabei umso größer, je weniger deutlich die Kompetenzen, Rechte und Pflichten der Partner untereinander abgegrenzt sind. Hier kann nur ein ausgewogener, unter Mitwirkung eines erfahrenen Rechtsanwaltes erstellter und von allen Partnern befürworteter Gesellschaftsvertrag Abhilfe schaffen. Dennoch bleibt ein Restpotential für Auseinandersetzungen, die, wenn sie nicht das Scheitern der Gründung verursachen, doch ein Wachstum des Unternehmens und seine Anpassung an den Markt erheblich behindern können.

Der Bereich der Finanzierung schließlich gestaltet sich für viele Existenzgründungen besonders problematisch. Strategische Entscheidungen sind hier vor allem über die aufzunehmenden Kredite zu treffen, die meist eine

1) Vgl. Szyperski, Nathusius (88), S.57.

Laufzeit von zehn Jahren oder mehr haben und somit eine langfristige Bindung bedeuten. Weiterhin besteht die Möglichkeit der Aufnahme von stillen Gesellschaftern, Kommanditisten oder Beteiligungen durch Beteiligungsgesellschaften (Venture Capital)[1].

Zusammenfassend muß festgestellt werden, daß sehr viele Gründer die strategische Dimension der angesprochenen, bei der Gründung zu treffenden Entscheidungen verkennen und diese daher nur ungenügend bzw. gar nicht planen. Es muß daher nach Meinung des Verfassers ein erstes Ziel sein, die Planungsbereitschaft der Gründer zu erhöhen und ihnen die nötigen Planungsinstrumente in für sie handhabbarer Form zur Verfügung zu stellen, wenn man die Erfolgsquote von Existenzgründungen erhöhen will.

4.1.4 Probleme der Existenzgründung

Untersuchungen zeigen, daß rund fünfzig Prozent der neugegründeten Unternehmen in den ersten fünf Jahren nach der Gründung liquidiert werden[2]. Als wesentliche Ursachen dafür können die unzureichende Ausstattung mit Eigenkapital, die schon bei kleineren Liquiditätsengpässen zur überschuldung führen kann, fehlende kaufmännische Kenntnisse und die mangelhafte Gründungsplanung, die zu Fehlentscheidungen in den strategischen Entscheidungsbereichen und somit zu einer langfristigen Beeinträchtigung des Unternehmenserfolges führt, gelten[3].

1) Vgl. Szyperski, Nathusius (88), S.57.
2) Vgl. ebenda, S.6.
3) Vgl. Bundesministerium für Wirtschaft (104), S.7 f.

Um den Eigenkapitalmangel der Jungunternehmen auszugleichen, hat die bundeseigene Deutsche Ausgleichsbank ein Eigenkapitalhilfe-Progamm[1] aufgelegt. Dieses Programm, das aus ERP-Mitteln[2] finanziert wird, soll die Gründung bei erfolgversprechenden Existenzgründungsvorhaben erleichtern. Dem Gründer wird ein langfristiger Kredit zu äußerst günstigen Konditionen zur Verfügung gestellt, für den außer der persönlichen Haftung des Gründers keine Sicherheiten erforderlich sind. Das dergestalt aufgenommene Kapital haftet gegenüber den Gläubigern wie Eigenkapital[3].

Neben dieser Unterstützung aus Bundesmitteln kommt eine Aufnahme von haftendem Kapital durch Beteiligungen in Frage. Viele Existenzgründer scheuen jedoch vor einer Aufnahme von Beteiligungskapital zurück, weil sie fürchten, dadurch in eine Abhängigkeit von den Beteiligungshaltern zu geraten[4]. Diese rational oft nicht zu begründende Scheu kann im Extremfall zum Scheitern des Gründungsprojektes führen, weil die zu niedrige Eigenkapitalquote nicht genug Finanzierungspielraum für die überbrückung von Liquiditätsengpässen durch Aufnahme zusätzlicher Kredite bietet.

Eine besondere Möglichkeit der Beteiligungsfinanzierung besteht in der Aufnahme von Wagniskapital (Venture Capital)[5]. Diese in den USA weit verbreitete Finanzierungsform beruht darauf, daß speziell für diesen Zweck gegründete Unternehmen Kapital für besonders erfolgversprechende Gründungen zur Verfügung stellen, wobei den Gründungsunternehmen oft eine relativ große unterneh-

---

1) Vgl. Bundesministerium für Wirtschaft (103), S.52 f.
2) ERP = "European Recovery Program"; vgl. Bundesministerium für Wirtschaft (102).
3) Vgl. Bundesministerium für Wirtschaft (103), S.52.
4) Vgl. Szyperski, Nathusius (88), S.57.
5) Vgl. ebenda.

merische Entscheidungsfreiheit gelassen und gleichzeitig Beratungsleistungen durch den Kapitalgeber zur Verfügung gestellt werden[1]. Allerdings beschränken sich die Aktivitäten dieser Gesellschaften in der Regel auf technologieorientierte Gründungen, so daß diese Finanzierungsform für den größten Teil der Existenzgründungen nicht in Frage kommt.

Der zweite Problemkreis, die mangelhafte Gründungsvorbereitung, ist vielschichtiger, aber vielleicht leichter zu beeinflußen. Möglicherweise hätte das Scheitern vieler Existenzgründungen, die eine unzureichende Kapitalausstattung hatten, durch eine bessere und realistischere Planung vermieden werden können. Letztlich geht es bei der Gründungsplanung auch darum, ggf. zu erkennen, daß eine Existenzgründung langfristig nicht tragfähig ist, und das Vorhaben in diesem Fall fallenzulassen.

Aus diesem Grund hat der Bund ein Beratungshilfe-Programm aufgelegt[2]. Aus diesem Programm werden 60% der Kosten für eine Gründungsberatung durch einen professionellen Berater bis zu einem Höchstbetrag von DM 2.500 erstattet. Bei der Gründungsberatung soll geklärt werden, ob das Gründungsvorhaben erfolgversprechend ist.

Zusätzlich stehen dem Gründer in den Kammern und Wirtschaftsförderungsämtern sowie teilweise in den Wirtschaftsverbänden Berater zur Verfügung, die ihn kostenlos oder gegen ein geringes Entgelt bei der Gründungsplanung unterstützen. Diese Stellen führen auch regelmäßig Seminare durch, in denen dem Gründer die wich-

1) Vgl. z.B. Techno Venture Management (113), Nord/Gl (109).
2) Vgl. Bundesministerium für Wirtschaft (103), S.64 ff.

tigsten betriebswirtschaftlichen und juristischen Sachverhalte der Existenzgründung vermittelt werden.

Trotz der genannten Maßnahmen kann festgestellt werden, daß die Gründungsvorbereitung nach wie vor unzureichend ist. Dies hat mehrere Gründe, von denen die wichtigsten hier kurz aufgeführt werden sollen:

Vielen Gründern mangelt es an betriebswirtschaftlichen Kenntnissen. Dieser Mangel läßt sich durch kurzfristige Fortbildungsseminare kaum hinreichend ausgleichen. Er könnte vielmehr durch eine stärkere Beachtung betriebswirtschaftlicher Fragen in der Berufsausbildung oder durch die Aufnahme betriebswirtschaftlich erfahrener Partner verringert werden[1].

Oft begründen Existenzgründer ihre mangelnde Planung damit, daß neben den zahlreichen Aufgaben und Verpflichtungen, die gerade in der Vorbereitungsphase anfielen, für strategische Planung keine Zeit mehr sei[2]. Dieses Argument kann zwar nicht ohne weiteres von der Hand gewiesen werden. Einiges spricht jedoch dafür, daß sich die Gründer über die Bedeutung der strategischen Gründungsplanung nicht im klaren sind und sich teilweise von Nebensächlichkeiten ablenken lassen.

Viele Gründer unterschätzen die notwendige Vorbereitungszeit und den erforderlichen Planungsaufwand. In dem Bestreben, ihre Selbständigkeit und Unabhängigkeit möglichst schnell zu erreichen, setzen sie sich selbst unter Zeitdruck und berauben sich damit der Möglichkeit einer sorgfältigen Auslotung der Gründungschancen und einer optimalen Abstimmung ihres Konzeptes auf den Markt. Die Erfolgschancen vieler Existenzgründungen

1) Vgl. Szyperski, Nathusius (88), S.40.
2) Vgl. ebenda, S.48.

könnten wesentlich erhöht werden, wenn der Gründer die Gründung langfristig plante und schon vorab, z.B. nebenberuflich oder in einer abhängigen Beschäftigung, erste Erfahrungen in dem angestrebten Markt sammelte.

- Mangelhafte Planung hat ihren Grund oft in fehlendem Wissen über Planungstechniken, die bei der Existenzgründung zu beachtenden Aspekte und möglichen Entscheidungsalternativen. So wissen z.B. viele Gründer von der Existenz der öffentlichen Finanzierungs- und Beratungshilfen nichts. Auch bei Finanzierungsberatungen durch die Banken werden diese nicht immer erwähnt, da die Vermittlung von öffentlichen Krediten für die Banken relativ unattraktiv ist. Hier kann eine verstärkte Aufklärung durch die öffentliche Hand helfen. Vor allem muß dem Gründer die Bedeutung einer gründlichen Vorinformation klargemacht werden.

Zusammenfassend kann gesagt werden, daß dem Existenzgründer vor allem durch verbesserte, auf sein Gründungsvorhaben abgestimmte Informationen und Planungshilfen geholfen werden kann. Von Politikern und öffentlichen Stellen wird immer wieder die Bedeutung von Existenzgründungen für die Konjunktur und Beschäftigungslage hervorgehoben. Gleichzeitig bedeutet eine gescheiterte Gründung einen erheblichen volkswirtschaftlichen Schaden. Hier könnte vor allem die Betriebswirtschaftslehre, die sich bisher kaum mit den speziellen Problemen der Existenzgründung befasst hat[1], einen wertvollen Beitrag leisten.

---

1) Vgl. Szyperski (87), S.151 f.

4.2 Die Konstruktion eines Expertensystems zur Existenzgründungs-Beratung

4.2.1 Aufgabe und Ziel eines Expertensystems zur Existenzgründungs-Beratung

Die Ausführungen in den vorhergehenden Abschnitten haben gezeigt, daß ein wesentliches Problem des Existenzgründers in mangelndem Wissen über die zu beachtenden Aspekte der Gründung liegt. Hier kann ein Expertensystem möglicherweise helfen, indem es dem Gründer das Wissen eines erfahrenen Existenzgründungs-Beraters zugänglich macht.

Aufgrund der Vielzahl der zu treffenden Entscheidungen und des Umfangs an notwendigen Informationen kann es jedoch nicht Aufgabe eines Expertensystems sein, dem Gründer ein vollständiges Gründungskonzept zu erstellen. Lediglich für Teilbereiche, etwa die Auswahl einer geeigneten Rechtsform, wäre eine sinnvolle Planungsunterstützung durch ein Expertensystem denkbar.

Stattdessen könnte ein Expertensystem jedoch ein durch den Gründer erstelltes Konzept prüfen und Mängel und Lücken in der Planung aufzeigen. Gleichzeitig kann es Informationen darüber zur Verfügung stellen, wie die gefundenen Lücken ausgefüllt werden können und an welchen Stellen weiterer Rat eingeholt werden kann. Legt man das Ablaufschema aus Abb. 4.1 zugrunde, so kann das Expertensystem bei der Durchführbarkeitsprüfung des in der zweiten Konkretisierungsphase erstellten Konzeptes helfen.

Ein derartiges System hätte eine ähnliche Funktion wie eine Checkliste zur Existenzgründungsplanung[1]. Gegen-

1) Solche Checklisten finden sich z.B. in Stück (85), Dittrich (17), Dresdner Bank (106).

über einer solchen Checkliste hätte es jedoch den Vorteil einer größeren Flexibilität und der Möglichkeit, vernetzte Strukturen und bedingte Abhängigkeiten zu berücksichtigen, die bei konventionellen Checklisten nicht oder nur mit erheblichem Aufwand zu berücksichtigen sind[1].

Als Zielaussage des Systems wäre dann die Fragestellung anzusehen, ob und welche Schwachstellen ein gegebenes Gründungskonzept enthält. Der Lösungsraum besteht damit im Prinzip nur aus der Aussage "Gründung ist sinnvoll" bzw. der Gegenaussage "Gründung ist nicht sinnvoll", die explizit in der Wissensbasis enthalten ist[2]. Betrachtet man den Zeitbezug des Problems, so ist sowohl eine rein statische, an der gegebenen Situation orientierte als auch eine dynamische, die mögliche zukünftige Entwicklung des Unternehmens berücksichtigende Betrachtungsweise möglich.

Das im folgenden beschriebene Expertensystem POUNDER bezieht die zukünftige Entwicklung durch ein einfaches Simulationsprogramm zur Liquiditätsprognose mit ein. Damit fällt das System in die Klasse der dynamischen Systeme mit explizit formuliertem Lösungsraum. Es handelt sich also nach der Klassifizierung in Abschnitt 3.3.1 um ein Interpretationsproblem.

Ordnet man das System in den realen Ablauf der Gründungsplanung ein, so könnte es als Vorfilter für ein Beratungsgespräch bei einem Berater der Kammern oder einem Unternehmensberater dienen. Der Gründer würde bereits vor einem solchen Gespräch auf Lücken in seiner Konzeption hingewiesen, so daß das Beratungsgespräch

1) Vgl. Mertens, Allgeyer, Däs (53), S.908, die "Intelligente Checklisten" als eigene Aufgabenklasse aufführen.
2) Vgl. Regel Nr.1 in Anhang A.

zielorientiert auf die Schwachstellen der Gründung gerichtet werden könnte. Auf diese Weise ließen sich die Effektivität und der Nutzen der Beratung wesentlich steigern und die Beratungskosten (die dem Gründer oder der zuständigen öffentlichen Stelle entstehen) senken.

Ein weiterer wesentlicher Nutzen des Systems wäre, daß der Gründer nicht nur auf Lücken in seinem Wissen hingewiesen wird, sondern gleichzeitig Informationen darüber erhält, wie und wo er sich das entsprechende Wissen beschaffen kann[1]. Schließlich lassen sich in ein derartiges System auch über die Prüfung des Konzeptes hinaus konkrete Vorschläge, z.B. über zu beantragende öffentliche Mittel, einarbeiten. Beides führt ebenfalls zu einer verbesserten Vorinformation, die die Gespräche mit Banken und öffentlichen Stellen effektiver machen und so den Gründer wesentlich entlasten kann.

1) Vgl. die Hinweistexte des Expertensystems POUNDER in Anhang A.

4.2.2 Die Struktur der Wissensbasis

Abb. 4.3 zeigt die Struktur der Wissensbasis des Expertensystems FOUNDER im überblick. Das System versucht herauszufinden, ob eine Existenzgründung auf der Basis eines vorliegenden Gründungskonzeptes sinnvoll ist und ggf. Schwachstellen aufzuzeigen. Dazu wird zunächst die zu bewertende Zielaussage "Gründung ist sinnvoll" in drei Teilaussagen aufgegliedert. Das System kommt zu einem positiven Gesamtergebnis, wenn alle diese Teilaussagen positiv bewertet werden.

Im ersten Teilbereich wird versucht herauszufinden, ob die Grundvoraussetzungen einer Existenzgründung erfüllt sind. Dazu werden einerseits die persönlichen Voraussetzungen des Gründers hinsichtlich körperlicher und geistiger Leistungsfähigkeit und seine Kenntnisse, andererseits die formalen Voraussetzungen geprüft.

Innerhalb des Unterpunktes "Personenbezogene Voraussetzungen" ermittelt das System abhängig von der angestrebten Branche, ob und welche speziellen Voraussetzungen und Genehmigungen vorliegen müssen, um eine Gründung juristisch zu ermöglichen. Der Punkt "Sachbezogene Voraussetzungen"[1] wird dagegen nicht weiter aufgegliedert, da die Fülle der Auflagen und Vorschriften, abhängig von der Art des Betriebes, innerhalb dieses Prototyp-Expertensystems nicht zu bewältigen war.

Der zweite Teilbereich prüft das Gründungskonzept daraufhin, ob die selbständige Existenz langfristig tragfähig ist. Dazu werden die Teilbereiche "Markt", "Unternehmung" und "Finanzierung" analysiert.

1) Hierzu zählen z.B. Auflagen betreffend den Umweltschutz, die gewerbliche Nutzung von Räumen etc.

Abb. 4.3: Struktur der Wissensbasis des Expertensystems FOUNDER

Innerhalb des Bereiches "Markt" wird geprüft, ob das Produkt bzw. die Dienstleistung marktgerecht ist und somit Grundlage eines langfristigen Erfolges sein kann. Dies umfaßt neben der Marktkenntnis des Gründers die aktuelle Marktsituation (Konkurrenz, Wachstumschancen), den Standort, den Beschaffungsmarkt, absatzpolitische Maßnahmen (Distribution und Werbung) sowie die "Unique Selling Proposition" (USP)[1], d.h. die Fähigkeit des Gründers, sich von der Konkurrenz abzugrenzen und sich dadurch Wettbewerbsvorteile zu verschaffen.

Im Bereich der "Unternehmung" spielen organisatorische Fragen wie die Rechtsform und die Aufnahme von Partnern eine Rolle. Die gewählte Rechtsform wird hier allerdings nicht weiter untersucht - dies wäre ein Ansatzpunkt für eine Erweiterung des Systems. Geprüft wird, ob der Gründer über die notwendigen Kenntnisse der Buchführung und Personalfragen verfügt, ob er sich über Alternativen zu seiner Gründungsform (z.B. Franchising) informiert hat und ob seine Planung ausreichend ist. Der letzte Punkt dient dazu, dem Gründer eindringlich die Bedeutung einer strategischen Gründungsplanung vor Augen zu führen. Ebenfalls von Bedeutung erscheint die Frage, ob der Gründer bereits Kontakt zu Fachberatern (Steuerberater, Rechtsanwalt) hat und somit in Spezialfragen einen Ansprechpartner kennt.

Im Teilbereich "Finanzierung" wird versucht herauszufinden, ob die geplante Finanzierung durchführbar ist und ausreichend Reserven enthält, um Liquiditätsschwierigkeiten zu vermeiden. Zunächst wird geprüft, ob die zur Deckung der notwendigen Kredite erforderlichen Sicherheiten vorhanden sind. Dann wird detailliert ge-

1) Vgl. Szyperski, Nathusius (88), S.40.

prüft, welche öffentlichen Mittel der Gründer beantragen kann. Entsprechende Hinweise mit Angabe der Kondi tionen werden ausgegeben. Weiterhin wird gefragt, ob der Gründer auch die notwendigen betrieblichen und pri vaten Versicherungen in der Planung berücksichtigt hat.

Schließlich wird in einem Simulationsmodell herauszufinden versucht, ob bei den gegebenen Erwartungen des Gründers hinsichtlich zukünftiger Ausgaben und Einnahmen die Liquidität des Betriebes gesichert ist. Dieses Modell ist als Prolog-Modul (intern definierte Relation) implementiert[1].

Der letzte Teilbereich umfaßt die Frage, ob eine selbständige Gründung für den Gründer überhaupt lohnend er scheint. Obwohl dies letztlich nur der Gründer selbst entscheiden kann, wird versucht, ihm die Vor- und vor allem Nachteile einer selbständigen Existenz vor Augen zu führen. Im Bereich der Motive wird geprüft, ob der Gründer genügend Motivation mitbringt, um die besondere Beanspruchung der Gründung durchzuhalten, und ob er bereit ist, auf viele seiner bisherigen Annehmlichkeiten zunächst zu verzichten.

Weiterhin wird das bisherige berufliche Umfeld untersucht. Dabei wird die selbständige Gründung als umso weniger lohnend eingestuft, je besser das Gehalt und die Aufstiegschancen innerhalb des bisherigen Arbeitsverhältnisses sind, vorausgesetzt, der Gründer ist nicht aus einem anderen Grund mit seinem bisherigen Arbeitsverhältnis unzufrieden. Dem Gründer wird klargemacht, daß er wesentlich mehr als sein bisheriges Bruttogehalt an Gewinn erwirtschaften muß, um seinen Lebensstandard aufrechterhalten zu können.

---

1) Vgl. die Abschnitte 2.5.5 und 4.2.5.

4.2.3 Die Wissensakquisition

Zur Erstellung der Wissensbasis des Expertensystems wurden verschiedene Literaturquellen herangezogen sowie eine Reihe von Experteninterviews durchgeführt. Das dabei verwendete schriftliche Material läßt sich unterteilen in wissenschaftliche Literatur[1], praxisbezogene Ratgeber[2] sowie Informationsmaterial öffentlicher und privater Stellen[3].

Die wissenschaftliche Literatur diente vor allem dazu, einen theoretischen Rahmen für die Problematik der Existenzgründungen zu erarbeiten und den Stand der Forschung auf diesem Gebiet beurteilen zu können. Dabei stellte sich heraus, daß sich aufgrund der wissenschaftlichen Literatur, selbst wenn sie empirisch orientiert war, unmittelbar noch keine Regeln für die Wissensbasis ableiten ließen[4]. Der Grund dafür dürfte in der wissenschaftlichen Sorgfalt liegen, die "Wenn-Dann-Aussagen" aufgrund ihres theoretisch kaum zu begründenden heuristischen Charakters verbietet.

Betrachtet man demgegenüber die praxisbezogenen Ratgeber, so enthalten diese bereits eine Fülle von Erfahrungen und Regeln, die sich für die Formulierung der Wissensbasis verwenden lassen. Dagegen werden kaum

---

1) Vgl. Szyperski, Nathusius (88); Szyperski (87); Rentrop (69); Wilhelm, Corsten, Peckedrahl (93).
2) Vgl. Dittrich (17), Hofmann (36), Honig (37), Kirschbaum, Naujoks (40), Rentrop (70), Stück (85), Wolf (96).
3) Vgl. Bundesministerium für Wirtschaft (102), (103), (104); Deutsche Ausgleichsbank (105); Dresdner Bank (106), (107); Industrie- und Handelskammer Nordrhein Westfalen (108); Norddeutsche Gesellschaft für Innovationsfinanzierung (109); Oldenburgische Landesbank (110); Sparkasse (111); Stadt Münster (112); Techno Venture Management (113).
4) Vgl. dazu Mertens, Allgeyer, Das (53), S.939.

theoretische Begründungen oder Nachweise für die Richtigkeit der angeführten Aussagen erbracht.

Die ausgewerteten Informationsmaterialien umfaßten weiterhin Merkblätter und Checklisten für Existenzgründer, die von privaten Banken oder öffentlichen Stellen herausgegeben wurden, sowie Informationsmaterial der zuständigen Stellen über die Vergabebedingungen und Konditionen öffentlicher Mittel. Gerade diese Materialien enthielten eine Fülle von Informationen, die direkt oder mit geringem Aufwand in die Wissensbasis übernommen werden konnten.

Die Auswertung schriftlicher Quellen für die Wissensbasis hat gezeigt, daß derartige Quellen in der Regel umso besser für eine Umsetzung in Regelwissen geeignet sind, je geringer ihr wissenschaftlicher bzw. theoretischer Anspruch ist. Dieses Phänomen dürfte dabei nicht nur auf den Bereich betriebswirtschaftlicher Probleme oder gar des hier verwendeten Beispiels beschränkt sein.

Bedenkt man, daß praxisbezogene, auf zahlreichen Heuristiken basierende Literatur oft von Experten geschrieben wird, die ihr Wissen nicht theoretisch begründen können, sondern sich auf ihre Berufserfahrung verlassen, so ist dieser Umstand nicht verwunderlich. Eben diese Form von Wissen ist es, die in Expertensystemen verwendet werden soll.

Nachdem eine Einarbeitung in die Problematik der Existenzgründung durch Studium einiger Literaturquellen erfolgt war, mußte ein Experte gefunden werden, der seine Erfahrungen über Probleme und Schwierigkeiten von Existenzgründern für die Wissensbasis zur Verfügung stellen konnte. Als Experte stellte sich freundlicher-

weise Herr Ltd.Gew.Dir.a.D. Dr. Reinhard Schneider, Osnabrück[1], zur Verfügung. Er war auch bei der Beschaffung ausführlichen Informationsmaterials für Existenzgründer bei einigen Privatbanken behilflich.

Herr Dr. Schneider verfügt über eine langjährige Erfahrung bei der Beratung von Existenzgründern. In einem ersten Interview wurde er gebeten, einige Fälle aus seiner Beratungspraxis darzustellen. Dabei wurde keine feste Form oder ein Ablauf vorgegeben. Es ging vielmehr darum, einen Einblick in die praktischen Probleme der Existenzgründung aus der Sicht eines unabhängigen Beraters zu erhalten und die bis dahin aus der Literatur noch recht undeutlich formulierten Vorstellungen zu präzisieren.

In der nächsten der jeweils ganztägigen Interviewsitzungen ging es darum, die bei einer Existenzgründung relevanten Gesichtspunkte zu formulieren. Aufgrund der Ergebnisse dieser Befragung wurde dann die Struktur der Wissensbasis, wie sie im vorigen Abschnitt dargestellt wurde, erarbeitet.

In den beiden folgenden Sitzungen wurden konkrete Sachverhalte und Regeln für eine erfolgreiche Existenzgründung erarbeitet. Dabei dienten immer wieder von Herrn Dr. Schneider dargestellte Fallbeispiele als Grundlage zur Ableitung von Bedingungen und Darstellung von Gefahren, die bei mangelhafter Vorbereitung auftreten können.

---

1) Herr Dr. Schneider ist Vorsitzender der "Senior-Experten Nordwest", einer Vereinigung von erfahrenen Senioren, die Jungunternehmer ehrenamtlich in Fragen der Existenzgründung und Unternehmenssicherung beraten. Er ist außerdem Mitglied des Sprechergremiums der Bundesarbeitsgemeinschaft der Senior-Experten, Bonn.

Zwischen den Interviewsitzungen wurde die Implementierung des neugewonnenen Wissens in einem Expertensystem mit Hilfe der Expertensystem-Shell MOSES vorgenommen. Dabei wurde auch auf die Literatur, sowohl wissenschaftliche als auch praxisbezogen-heuristische bzw. Informationsmaterial, zurückgegriffen, um das bei den Interviews gesammelte Wissen zu überprüfen und zu ergänzen.

Die fünfte und sechste Interviewsitzung schließlich dienten der überprüfung des entstandenen Expertensystems. Dazu wurde das Expertensystem auf ein hypothetisches Fallbeispiel angewandt. Nach der ersten überprüfung stellte sich heraus, daß eine Vertiefung der Wissensbasis in den Bereichen "formale personenbezogene Voraussetzungen" und "öffentliche Mittel" wünschenswert erschien. Daraufhin wurde die Befragung noch einmal auf diese Punkte konzentriert.

Nach der sechsten Interviewsitzung wurde die Wissensakquisition schließlich als abgeschlossen angesehen, nachdem keine gravierenden Mängel mehr in der Wissensbasis gefunden wurden. Das resultierende Expertensystem kann als funktionsfähiger Demonstrations-Prototyp angesehen werden. Der Umfang der Wissensbasis kann mit mehr als 100 Regeln als für die Zwecke dieser Arbeit ausreichend bezeichnet werden. Für den praktischen Einsatz erscheint jedoch eine Erweiterung in einigen Teilbereichen sowie eine extensive Validierung der Wissensbasis, möglichst unter Hinzuziehung weiterer Experten, notwendig.

Nach Ansicht des Verfassers haben die Erfahrungen bei der Wissensakquisition für das Expertensystem FOUNDER deutlich gezeigt, daß auf die Mithilfe eines erfahrenen Experten auch bei der Konstruktion von Versuchs- und Demonstrationssystemen nicht verzichtet werden sollte.

Es zeigte sich immer wieder, daß das Umsetzen des Erfahrungswissens des Experten in Regeln einer Wissensbasis die anspruchsvollste Aufgabe des Knowledge Engineering ist. Gerade in diesem Punkt konnten wertvolle Erfahrungen gesammelt werden, die in die Ausführungen in Kapitel 3 eingearbeitet wurden.

Wesentlich für den Erfolg des Projektes war die Aufgeschlossenheit und Bereitschaft zur Mitarbeit von Herrn Dr. Schneider. Voraussetzung dafür war, daß er den Fortschritt der Entwicklung unmittelbar beobachten konnte, daß eine entspannte und produktive Gesprächsatmosphäre geschaffen werden konnte und daß er ausreichend Zeit für die Befragungen zur Verfügung stellen konnte.

Bei den Experteninterviews wurde ein großer Teil der in Abschnitt 3.1.3 beschriebenen Methoden eingesetzt. Dabei wurde keinem vorher festgelegten Schema gefolgt; lediglich der Themenbereich und das Befragungsziel wurden für jede Interviewsitzung zuvor definiert. Es zeigte sich, daß ein ständiger Wechsel zwischen beobachtender und intuitiver Methode[1] zu einer effizienten Form der Befragung führte.

Zu Beginn wurde Herr Dr. Schneider z.B. gebeten, einen Fall aus der Praxis darzustellen. Zu diesem Fall wurden dann vom Verfasser Fragen gestellt und Hypothesen, z.B. über Gründungsfehler, entwickelt und mit dem Experten diskutiert. Danach wurde versucht, ein weiteres praktisches Fallbeispiel zu finden, anhand dessen die entwickelten Hypothesen bestätigt oder verworfen werden konnten, oder es wurde das vorherige Fallbeispiel variiert, um die Auswirkungen einer veränderten Gründungsstrategie zu diskutieren.

1) Vgl. Abschnitt 3.1.2.

Die Erfahrung bei den Experteninterviews hat gezeigt, daß eine wesentliche Schwierigkeit zunächst darin besteht, eine gemeinsame Kommunikationsgrundlage zwischen Knowledge Engineer und Experte herzustellen. Dies bedeutet, daß der Knowledge Engineer lernen muß, sich in die Denkweise des Experten hineinzuversetzen. Der Experte wiederum muß zunächst lernen, worauf es bei der Formulierung von Wissen für ein Expertensystem ankommt, so daß die Befragung zielstrebiger auf die Kernpunkte des Problems hinsteuern kann. Im vorliegenden Fall konnte diese gemeinsame Kommunikationsgrundlage innerhalb relativ kurzer Zeit geschaffen werden.

4.2.4 Erläuterungen der Regeln

Anhang A enthält die Wissensbasis des Expertensystems FOUNDER. Sie ist untergliedert in die Abschnitte "Titelseite", "Regeln", "Fragetexte" und "Hinweistexte"[1]. Die Titelseite enthält allgemeine Erläuterungen für den Benutzer, die nach dem Laden der Wissensbasis angezeigt werden. Die Fragetexte sind für einzelne Aussagen definiert und werden als Frage an den Benutzer ausgegeben, wenn der Wahrheitswert der entsprechenden Aussage beurteilt werden soll. Hinweistexte werden dagegen ausgegeben, nachdem der Wahrheitswert einer Aussage im angegebenen Bereich bestätigt werden konnte.

Die Wissensbasis wurde mit Hilfe eines speziell dafür erstellten Programmes ausgedruckt, das hinter jeder Aussage eines Regelkörpers anzeigt, welche Regeln diese Aussage im Regelkopf enthalten, bzw. ob ein Fragetext für diese Aussage definiert ist. Auf diese Weise läßt sich der Lösungsweg einer Aussage durch die Regelbasis bequem verfolgen.

1) Vgl. Abschnitt 2.5.

Im folgenden soll die Bedeutung einzelner Regeln näher erläutert werden. Auf die Frage- und Hinweistexte wird dagegen nicht näher eingegangen. Man sollte jedoch beachten, daß gerade die Hinweistexte ein wichtiges Element der Wissensbasis sind, da sie dem Benutzer Hilfestellungen und Informationen vermitteln können.

Die Regeln 1 - 10 gliedern die Wissensbasis hierarchisch entsprechend dem Schema in Abb. 4.3. Während Regel 1 die ursprüngliche Zielaussage in drei Teilaussagen spaltet, konkretisieren die Regeln 2, 5 und 9 diese Teilaussagen weiter. Die Regeln 3, 4, 6, 7, 8 und 10 schließlich bilden eine dritte Konkretisierungsebene. Kennzeichnend für diese Strukturierung ist, daß sämtliche in den Regeln 1 - 10 enthaltenen Aussagen für die Zielaussage kritisch ("Und-verknüpft") sind, d.h. die Zielaussage wird mit dem Minimum der Wahrheitswerte aller dieser Aussagen bewertet.

Ebenfalls typisch für Regeln, die der Strukturierung der Wissensbasis dienen, ist es, daß alle Regeln den Glaubwürdigkeitswert 1 (immer zutreffend) erhalten. Anstelle der Regeln 1 - 10 hätte man auch alle Aussagen auf der dritten Konkretisierungsstufe in einer einzigen Regel zusammenfassen können. Dies hätte jedoch zu einer unübersichtlichen und schlecht nachvollziehbaren Wissensbasis geführt.

Der Hinweis "Keine Referenz!" hinter der Bedingung "die Liquidität_ist gesichert" in Regel 8 bedeutet, daß weder eine Regel noch ein Fragetext für diese Aussage existiert. Dies ist oft ein Hinweis auf einen Fehler, der z.B. durch eine nicht übereinstimmende Schreibweise bei Bedingung und Regelkopf auftreten kann. In diesem Fall jedoch wird die Aussage "die Liquidität_ist gesichert" durch ein PROLOG-Unterprogramm intern bewertet. Da Informationen über intern definierte Relationen

nicht mit der Wissensbasis abgespeichert werden, kann das Druckprogramm diesen Umstand nicht erkennen.

Die Regeln 11 und 12 untersuchen, ob die berufliche Situation des Gründers gegen eine Gründung spricht. Dies wird dann als gegeben angenommen, wenn der Gründer mit seinem Beruf zufrieden ist und er entweder ein seinen Fähigkeiten entsprechend überdurchschnittliches Gehalt erhält oder in seinem Betrieb für ihn gute Aufstiegschancen bestehen. In diesen Fällen wird der Gründer durch die Gründung vermutlich kaum Vorteile erlangen können, aber erhebliche Risiken in Kauf nehmen müssen.

Die Regeln 13 und 14 untersuchen die körperliche Leistungsfähigkeit des Gründers. Es wird angenommen, daß sie gegeben ist, wenn der Gründer regelmäßig Sport treibt oder wenn er sich "fit" fühlt und keine chronischen Leiden bekannt sind. Für diese Aussagen sind Fragetexte definiert. Es wird der höhere der aus beiden Regeln ermittelten Wahrheitswerte an die Regel 3 weitergegeben.

Die Regeln 15 und 16 befassen sich mit der Frage, ob die Familie des Gründers bereit ist, ihn bei der Gründung zu unterstützen. Dies ist nach praktischen Erfahrungen ein wichtiges Element der Gründung, da der Gründer durch die starken Belastungen während der Gründungsphase verstärkt auf die Unterstützung der Familie angewiesen ist. Es wird danach gefragt, ob der Gründer bereits mit seiner Familie gesprochen hat und diese dem Vorhaben zugestimmt hat.

Für den Fall, daß der Gründer keine Familie hat, wird die "Unterstützung" durch die Familie als gegeben angenommen, da in diesem Fall zumindest keine zusätzlichen Belastungen von Seiten der Familie zu erwarten sind. Die Bedingung "Gründer hat Familie" in Regel 16 dient

dazu, zu verhindern, daß nach der Unterstützungsbereitschaft der Familie gefragt wird, obwohl der Gründer bereits in Regel 15 geantwortet hat, er habe keine Familie.

Die Regeln 17 - 21 befassen sich mit der geistigen Belastbarkeit und den notwendigen Kenntnissen des Gründers. Dabei werden die Fachkenntnisse als gegeben angesehen, wenn der Gründer eine fachbezogene Berufsausbildung oder eine mehrjährige Branchenerfahrung hat. Die kaufmännischen Kenntnisse werden als gegeben angesehen, wenn der Gründer eine entsprechende Berufsausbildung hat oder wenn er selbst glaubt, ausreichende Kenntnisse zu haben.

Regel 21 macht deutlich, daß viele Fragen des Systems in erster Linie dazu dienen, den Gründer auf eigene Schwächen bzw. Schwächen des Konzeptes aufmerksam zu machen, und weniger das Ziel haben, eine objektive Prognose des späteren Unternehmenserfolges zu geben.

Die Regeln 22 und 23 überprüfen die formalen, personenbezogenen Voraussetzungen für den Fall der Gründung eines handwerklichen Betriebes. In diesem Fall muß der Gründer selbst Handwerksmeister und in die Handwerksrolle eingetragen sein oder er muß einen Handwerksmeister einstellen, der für die Ausführung der Handwerksarbeiten verantwortlich ist. Nach Regel 24 soll für den Fall, daß kein Handwerksbetrieb gegründet werden soll, ermittelt werden, welche Bestimmungen der Gründer im einzelnen zu beachten hat (Regeln 68 - 90, s.u.).

Die Regeln 25 - 42 enthalten Wissen darüber, unter welchen Bedingungen der Markt für die Produkte oder Dienstleistungen des Gründers erfolgversprechend erscheint. Dabei befassen sich die Regeln 25 und 26 mit der Frage, ob die Marktkenntnisse des Gründers ausreichend sind. Dies wird als gegeben vorausgesetzt, wenn

der Gründer eine mehrjährige Branchenerfahrung besitzt oder wenn er sich bereits soweit mit dem Markt befasst hat, daß er die speziellen Erfolgsfaktoren (Critical Success Factors)[1] und die führenden Konkurrenten zu kennen glaubt.

Die Regeln 27 - 29 sollen prüfen, ob ein ausreichendes Marktpotential[2] vorhanden ist. Dies wird als gegeben angenommen, wenn für das Produkt ein konkreter und ausreichender Bedarf vorhanden ist und der Markt noch nicht gesättigt ist bzw. das Produkt innovativ ist. Außerdem wird gefragt, ob bereits eine Marktanalyse durch einen unabhängigen Gutachter durchgeführt wurde.

Die Regeln 30 - 33 fragen nach der "Unique Selling Proposition (USP)"[3], die es ermöglichen soll, sich von der Konkurrenz abzuheben und damit Kundenpräferenzen schaffen zu können. Eine ausreichende USP wird angenommen, wenn das Produkt bzw. die Dienstleistung bei gegebener Qualität besonders preiswert angeboten werden kann (Preis-Leistungs-Verhältnis), wenn es innovativ (d.h. in dieser Form noch nicht auf dem Markt) ist bzw. in eine Marktlücke stößt oder wenn die Qualität deutlich höher als die der Mitbewerber ist.

Die Regeln 34 - 36 untersuchen die Konkurrenzaktivität im angestrebten Markt. Dabei wird unterstellt, daß eine Gründung in einem hart umkämpften Markt nicht sinnvoll ist. Ein Indiz für einen derartigen Markt ist es, wenn kleinere Unternehmen der Branche in der Vergangenheit häufiger schließen mußten (weil sie dem harten Konkurrenzkampf nicht gewachsen waren). Ebenso wird der Markt

---

1) Vgl. Rentrop (69), S.47 ff.
2) Dies bezieht sich auf das Marktpotential des gesamten Marktes. Das standortabhängige Marktpotential (Einzugsgebiet) wird in Regel 38 geprüft.
3) Vgl. Szyperski, Nathusius (88), S.40.

als ungeeignet eingestuft, wenn ein Großunternehmen in genau demselben Bereich wie der Gründer aktiv ist, da dieses vermutlich Markteintrittsbarrieren errichten oder Abwehrmaßnahmen ergreifen wird, gegen die der Gründer sich bei seiner geringen Kapitalausstattung nicht wehren kann.

Die Regeln 37 und 38 fragen nach der Eignung des geplanten Standortes. Dieser wird als geeignet angesehen, wenn von einem unabhängigen Gutachter eine Standortanalyse durchgeführt wurde und diese positiv ausgefallen ist[1]. Andernfalls werden eine Reihe von Bedingungen geprüft, die die Verkehrsanbindung, die Kundschaft, das betriebliche Umfeld, das Einzugsgebiet, die Konkurrenzaktivität und die Standortkosten (Miete etc.) umfassen.

Durch die Frage nach der künftigen Entwicklung des Standortumfeldes soll sichergestellt werden, daß der Gründer sich z.B. bei der Stadtverwaltung nach Plänen für Neu- und Umgestaltungen erkundigt hat, die den Geschäftserfolg nachhaltig beeinflußen können. Weiterhin wird der Gründer danach gefragt, ob er wenigstens einige weitere potentielle Standorte untersucht hat, bevor er sich entschieden hat. Ist dies nicht der Fall, so kann mit hoher Wahrscheinlichkeit davon ausgegangen werden, daß der gewählte Standort nicht der günstigste ist[2].

Die Fragen 39 - 42 befassen sich mit der Beschaffung, Distribution (die nur bei Produktionsbetrieben eine Rolle spielt) und Werbung. Im Bereich der Beschaffung wird untersucht, ob eine einseitige Abhängigkeit von einem Lieferanten droht, die die dringend nötige Flexibilität des Gründers erheblich einschränken kann. Die

1) Standortanalysen können z.B. von der örtlichen Industrie- und Handelskammer durchgeführt werden.
2) Vgl. Abschnitt 4.1.3.

Distribution wird als gewährleistet angesehen, wenn der Gründer einen Vertriebsweg ausgewählt und sichergestellt hat, daß dieser auch für ihn zugänglich ist. Im Bereich der Werbung wird erwartet, daß der Gründer bereits eine grobe Werbestrategie entwickelt hat und die Bedeutung der Werbung für den Geschäftserfolg nicht unterschätzt[1].

Die Regeln 43 - 60 befassen sich mit den organisatorischen Fragen der Unternehmung. Zunächst wird in Regel 43 die Frage nach der Rechtsform angeschnitten, wobei aber lediglich geprüft wird, ob der Gründer sich darüber bereits Gedanken gemacht und diese mit einem Steuerberater abgestimmt hat. Für eine Erweiterung des Systems böte sich an, an dieser Stelle zu prüfen, welche Rechtsform für den Gründer am besten geeignet ist.

Regel 44 behandelt die wichtige Frage nach eventuellen Geschäftspartnern. Dabei wird neben einem Vertrauensverhältnis zwischen den Partnern auch eine klare Regelung der Kompetenzen für wichtig erachtet, um späteren Streitigkeiten vorzubeugen. Natürlich müssen auch die Geschäftspartner die in den Regeln 13 ff. angeführten persönlichen Voraussetzungen erfüllen. Schließlich sollte darauf geachtet werden, daß die Belastungen und Erträge der Gründung gleichmäßig auf alle Beteiligten aufgeteilt sind.

Die Regeln 45 - 48 befassen sich mit der Gründungsplanung, deren Bedeutung hier bereits mehrfach hervorgehoben wurde. Für eine ausreichende Gründungsplanung werden als Teilpläne ein Finanzierungsplan, ein Liquiditätsplan und ein Marketingplan benötigt.

---

1) Vgl. Regel 67.

Der Gründer sollte die Bedeutung der Planung nicht unterschätzen. Um dies zu prüfen, wird ihm in Regel 46 die Aussage "Planung kostet viel Zeit und bringt wenig. Letztlich kann man ja doch nicht wissen, wie´s kommt." vorgelegt. Stimmt der Gründer dieser Aussage zu, so ist seine Planung wahrscheinlich unzureichend. Mit einem Hinweistext wird er auf die Bedeutung der Planung für eine erfolgreiche Gründung aufmerksam gemacht.

Die Planung muß nicht nur die erforderlichen Teilpläne umfassen, sondern sie sollte hinreichend detailliert und realistisch sein. Um dies zu beurteilen, wird der Gründer gefragt, ob er die voraussichtliche Höhe der Betriebsausgaben im achten Monat nach Gründung und den genauen Umsatz im September des dritten Jahres nach Gründung vorhersagen kann. Beantwortet er die erste Frage negativ, so ist seine Planung vermutlich nicht detailliert genug, um eine sinnvolle Liquiditätsplanung zu gewährleisten. Beantwortet er dagegen die zweite Frage positiv, so ist seine Planung sicherlich "blauäugig", d.h. naiv und unrealistisch.

Regel 49 befasst sich mit der Frage, ob der Gründer auf dem neuesten technischen Stand seiner Branche ist und somit nicht Gefahr läuft, durch veraltete Methoden bzw. Produkte in Kosten- oder Absatznachteile zur Konkurrenz zu geraten.

Regel 52[1] untersucht, ob der Gründer auch Alternativen zu seiner geplanten Gründungsform geprüft hat. Alternativen zu einer selbständigen, unabhängigen Neugründung wären z.B. die übernahme eines bestehenden Betriebes, die Errichtung eines Franchise-Betriebes, der gerade jungen Gründern oft eine gute Chance für einen selb-

1) Die Regeln 50 und 51 wurden im Zuge einer überarbeitung der Wissensbasis ersatzlos entfernt.

ständigen Start als Unternehmer ermöglicht, oder eine Kooperation mit anderen Gründern oder bereits bestehenden Betrieben.

Die Regeln 53 und 54 fragen danach, ob der Gründer sich ausreichend mit Personalfragen befasst hat. Dabei kommt es nicht nur darauf an, die bestehenden Vorschriften (z.B. hinsichtlich Arbeitsstättenverordnung, Kündigungsschutz usw.) zu kennen, sondern auch sicherzustellen, daß überhaupt das benötigte, hinreichend qualifizierte Personal verfügbar ist. Dieses von Gründern oft übersehene Problem entsteht daraus, daß qualifizierte Fachkräfte das Risiko einer Anstellung in einem neugegründeten Unternehmen scheuen und der Gründer zudem oft nicht mit den Gehältern der etablierten Unternehmen konkurrieren kann[1]. Sofern kein Personal für die Gründung benötigt wird, gilt diese Frage als erledigt.

Die Regeln 55 und 56 prüfen, ob die Beratung durch unabhängige, qualifizierte Fachleute während der Gründungsphase gesichert ist. Dies ist der Fall, wenn der Gründer bereits Kontakt zu einem Steuerberater und Rechtsanwalt hat und er die Bedeutung einer unabhängigen Beratung richtig einschätzt. Um dies zu prüfen, wird ihm ähnlich wie in Regel 46 eine Aussage zur Beurteilung vorgelegt, die zur Beratung durch externe Berater negativ Stellung nimmt.

Stimmt der Gründer dieser Aussage zu, so besteht die Gefahr, daß er später eine rechtzeitige Einholung fachkundigen Rates versäumt und sich erst dann an externe Berater wendet, wenn diese ihm nicht mehr helfen können. Dieses Phänomen konnte z.B. Herr Dr. Schneider in seiner Beratungspraxis oft beobachten.

1) Vgl. Szyperski, Nathusius (88), S.59.

In den Regeln 57 - 60 wird geprüft, ob eine ausreichende Buchführung durch den Gründer oder einen angestellten oder externen Buchhalter (sog. mobiler Buchführungshelfer) gewährleistet ist. Ist dies nicht der Fall, so wird der Gründer durch einen Hinweistext auf die Bedeutung der Buchführung als Kontroll und Informationsinstrument aufmerksam gemacht.

Die Regeln 61 - 65 prüfen die Finanzierung des Gründungsvorhabens. Dabei untersuchen zunächst die Regeln 61 und 62, ob ausreichende Sicherheiten für die Vergabe der benötigten Kredite vorhanden sind. Dies ist der Fall, wenn der Gründer über entsprechende bankübliche Sicherheiten (z.B. eine Bürgschaft, ein Grundstück oder Haus) verfügt oder wenn ihm ein Kreditberater seiner Bank eine positive Zusage gemacht hat, z.B. weil ein Gründungserfolg aus der Sicht der Bank wahrscheinlich ist.

Regel 63 prüft, welche verschiedenen Finanzierungs- bzw. Beratungshilfen der öffentlichen Hand für den Gründer in Frage kommen[1]. Zuvor wird jedoch gefragt, ob der Gründer eine entsprechende Beratung überhaupt wünscht. Ist dies nicht der Fall, so wird eine Ausschöpfung aller in Frage kommenden öffentlichen Mittel als gegeben angesehen[2]. Damit läßt sich vermeiden, daß detailliert geprüft wird, welche öffentlichen Mittel für den Gründer in Frage kommen, obwohl dieser auf diesem Gebiet gar keinen Informationsbedarf hat.

---

1) Dies wird detailliert in den Regeln 92 - 111 untersucht.
2) Vgl. Regel 91. Das Beispiel zeigt, daß auch die Tiefe der Analysen eines Expertensystems vom Benutzer gesteuert werden kann. Auf diese Weise läßt sich unter Umständen die Benutzerfreundlichkeit und Akzeptanz des Systems erhöhen.

Regel 64 geht noch einmal auf die Finanzplanung ein, wobei hier geprüft wird, ob der Gründer die "Durststrecke" der ersten sechs bis zwölf Monate nach der Gründung angemessen berücksichtigt und auch eine Reserve für unvorhergesehene Ausgaben oder Umsatzeinbrüche einkalkuliert hat. Ebenso muß der Gründer beachten, daß durch Zahlungsverzug der Abnehmer gerade in der Anfangszeit gefährliche Liquiditätsengpässe auftreten können.

Regel 65 prüft die sog. "goldene Bilanzregel", die besagt, daß das langfristige Kapital das langfristige Vermögen decken bzw. die Laufzeit des Finanzierungsmittels der Nutzungsdauer des finanzierten Anlagegutes entsprechen soll. Diese alte, theoretisch umstrittene, aber in der Praxis weit verbreitete Heuristik ist ein typisches Beispiel für eine in einem Expertensystem verwendete "Faustregel".

Regel 66 definiert, daß ein Industriebetrieb gegründet werden soll, wenn der Gründer dies angibt und nicht zuvor bereits gesagt wurde, daß ein Handwerksbetrieb gegründet werden soll. Diese Bedingung führt dazu, daß die Frage nach dem Industriebetrieb nur gestellt wird, wenn zu diesem Zeitpunkt noch unklar ist, welche Betriebsform gewählt wird. Dabei ist die Reihenfolge der Bedingungen im Regelkörper entscheidend. Wird die Aussage "Betrieb ist Handwerk" mit 1 bewertet, so bricht der Inferenzmechanismus an dieser Stelle ab, da die Zielaussage "Betrieb ist Industrie" nicht mehr besser als 0 bewertet werden kann. Eine derartige, sich gegenseitig ausschließende Verknüpfung mehrerer Regeln mit gleicher Zielaussage ist bereits mehrfach in der Wissensbasis verwendet worden[1].

1) Vgl. z.B. die Regeln 22 24.

Regel 67 geht noch einmal[1] auf die Einschätzung der Werbung durch den Gründer ein, indem ihm ähnlich wie im Bereich der Planung und Beratung eine Aussage vorgelegt wird, die die Bedeutung der Werbung negativ beurteilt. Daß die Regel soweit "abgelegen" in der Wissensbasis auftaucht, liegt darin begründet, daß sie beim ersten Erstellen der Regeln vergessen wurde. Die Expertensystem-Shell MOSES verfügt nicht über die Möglichkeit, Regeln nachträglich an anderen Stellen der Wissensbasis einzufügen. Dies hat jedoch keinen Einfluß auf den logischen Ablauf der Schlußfolgerungen, sondern nur auf die übersichtlichkeit der Wissensbasis.

Die Regeln 68 - 90 prüfen, welche speziellen Genehmigungen der öffentlichen Stellen zur Gründung vorliegen müssen und ob diese erteilt worden sind. Geprüft werden im einzelnen die Reisegewerbekarte, die vorliegen muß, wenn der Gründer keinen festen Standort anstrebt oder einen wesentlichen Teil seines Umsatzes nicht an seinem Standort hat, und der Nachweis der Sachkunde oder eine besondere Genehmigung, die für den Handel mit bestimmten Warengruppen oder bestimmte Dienstleistungen erforderlich ist. Auf die einzelnen Regeln soll hier nicht mehr näher eingegangen werden.

Die Regeln 91 - 111 schließlich prüfen die in Frage kommenden öffentlichen Mittel anhand ihrer Vergabebedingungen. Geprüft werden im einzelnen das Eigenkapitalhilfe-Programm, das ERP-Existenzgründungsprogramm und die Ergänzungsprogramme I und II der Deutschen Ausgleichsbank, die Bürgschaften der Kreditgarantiegemeinschaften sowie die Beratungsförderung des Bundes für Existenzgründer[2].

---

1) Vgl. Regel 42.
2) Vgl. Bundesministerium für Wirtschaft (103), S.52 ff.

Die Finanzierungshilfen der einzelnen Bundesländer werden nicht geprüft, da dies den Rahmen dieses Prototyp-Systems sprengen würde. Stattdessen wird ein Hinweis darauf gegeben, daß eventuell zusätzlich Förderprogramme der Länder in Frage kommen und der Gründer sich entsprechend informieren sollte.

4.2.5 Einbindung eines Simulationsmodells zur Liquiditätsprognose

Um die Möglichkeit, konventionelle Problemlösungstechniken in Expertensysteme einzubeziehen, zu demonstrieren, wurde ein Simulationsmodell[1] zur Liquiditätsprognose als PROLOG-Unterprogramm entwickelt und als MOSES-Modul in das Expertensystem FOUNDER eingebunden[2]. Dazu wurde die Relation "Liquidität_ist" als "intern" deklariert. Anhang C enthält das PROLOG-Listing des Moduls.

Wenn die MOSES-Schlußfolgerungskomponente Regel Nr.8 abarbeitet, stößt sie dabei auf die Aussage "die Liquidität ist gesichert". Sie prüft, ob die Relation "Liquidität_ist" intern definiert ist und stellt fest, daß dies der Fall ist. Nun ruft sie das PROLOG-Prädikat "ist_wahr" mit den drei Komponenten der Aussage als Eingabeparameter auf. Dieses Prädikat ruft seinerseits weitere PROLOG Prädikate auf, die die Bestandteile des Simulationsmodells bilden. Schließlich übergibt die Relation "ist_wahr" als Ausgabeparameter den Wahrheitswert für die Zielaussage, der auf der Basis des Simulationsergebnisses ermittelt wurde.

Um zu beurteilen, wie groß die Wahrscheinlichkeit einer Illiquidität in den ersten fünf Jahren nach der Gründung ist, werden 100 Simulationsläufe durchgeführt.

---

1) Eine kurze Einführung in die Simulationstechnik gibt z.B. Müller-Merbach in (58), S.451 ff.
2) Vgl. Abschnitt 2.5.5.

Abb. 4.4 zeigt den Ablaufplan des Simulationsmodells. Eingabeparameter sind die geschätzten Umsatz- und Kostengrößen der ersten fünf Jahre, eine evtl. Liquiditätsreserve zu Beginn und ein als konstant angesehener Prozentsatz als Anteil der variablen Kosten (z.B. Materialkosten, Wareneinsatz) am Umsatz.

Dabei werden für den Umsatz pro Jahr drei verschiedene Werte benötigt, die der Gründer schätzen muß: einen schlechtesten Umsatz, den er erwartet, wenn ein besonders schlechtes Jahr vorliegt, einen besten Umsatz, der einem besonders guten Jahr entspricht, und den Umsatz, den er als den wahrscheinlichsten annimmt. Aus diesen Werten wird für jedes Jahr eine Dreiecksverteilung[1] berechnet, anhand derer bei jedem Simulationslauf ein Umsatzwert zufällig ermittelt wird. Dabei ist die Wahrscheinlichkeit, daß der Umsatz einen bestimmten Wert annimmt, umso größer, je dichter dieser am vom Gründer als wahrscheinlich angesehenen Wert liegt.

---

1) Der dreiecksverteilte Wert U (z) in Abhängigkeit von der im Intervall 0 <= z <= 1 gleichverteilten Zufallszahl z berechnet sich nach den Formeln

$$U(z) = UG + ((EW - UG) * (OG - UG) * z))^{0.5}$$

für $z <= z_c$ bzw.

$$U(z) = OG - ((OG - EW)^2 \quad (OG - EW) * (OG - UG) * (z - z_c))^{0.5}$$

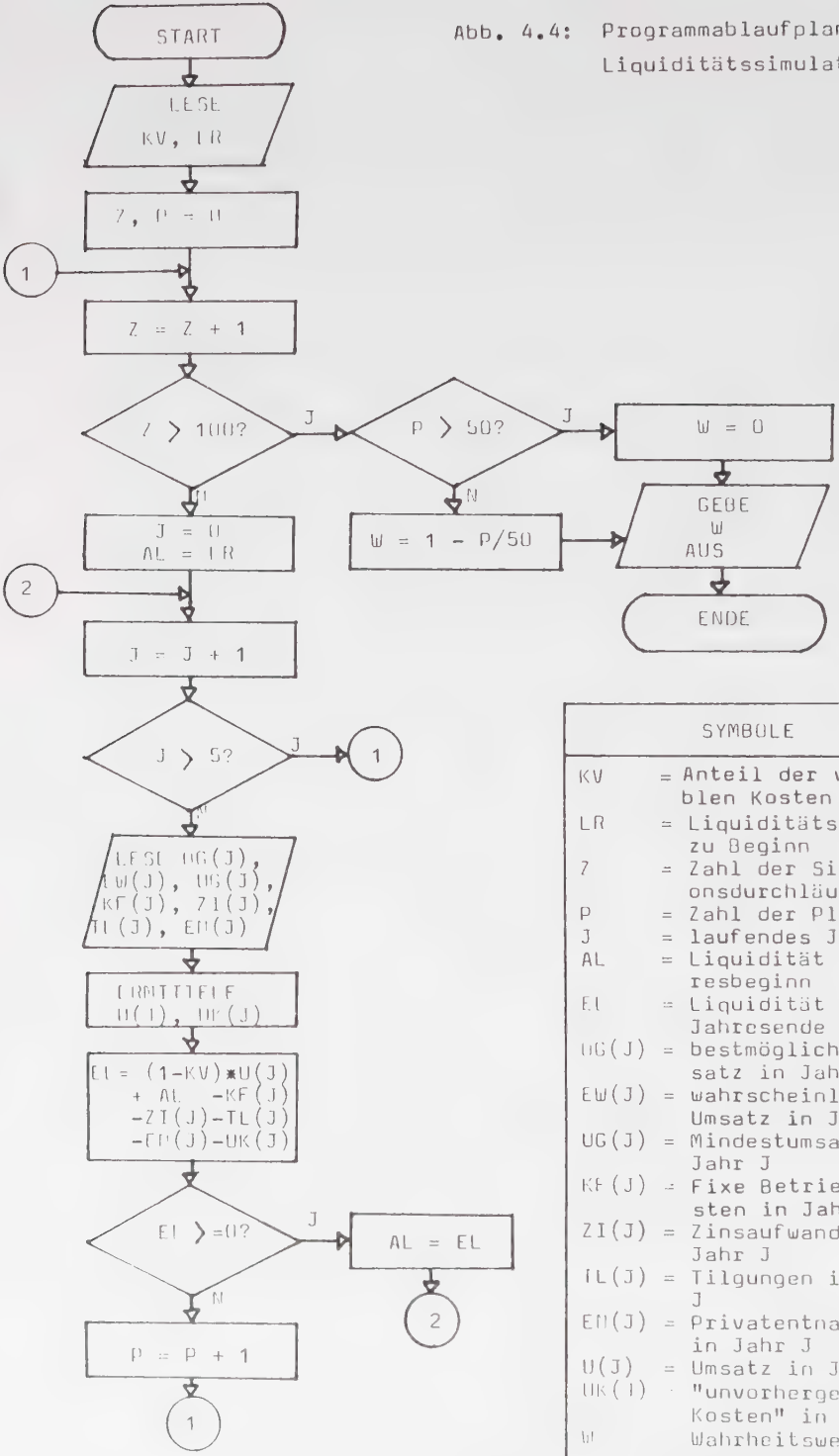
für $z >= z_c$, wobei gilt

$$z_c = (EW - UG) / (OG - UG).$$

Dabei bedeutet

- U (z) = zufallsabhängiger Umsatz
- OG = bestmöglicher Umsatz
- EW = wahrscheinlichster Umsatz
- UG = schlechtester Umsatz
- z = im Intervall 0..1 gleichverteilte Zufalls variable
- z_c = "kritischer Wert" für z, wobei gilt: $U(z_c) = EW$.

Abb. 4.4: Programmablaufplan zur Liquiditätssimulation



SYMBOLE	
KV	= Anteil der variablen Kosten an U(J)
LR	= Liquiditätsreserve zu Beginn
Z	= Zahl der Simulationsdurchläufe
P	= Zahl der Pleiten
J	= laufendes Jahr
AL	= Liquidität zu Jahresbeginn
EL	= Liquidität zum Jahresende
OG(J)	= bestmöglicher Umsatz in Jahr J
EW(J)	= wahrscheinlichster Umsatz in Jahr J
UG(J)	= Mindestumsatz in Jahr J
KF(J)	= Fixe Betriebskosten in Jahr J
ZI(J)	= Zinsaufwand in Jahr J
TL(J)	= Tilgungen in Jahr J
EN(J)	= Privatentnahmen in Jahr J
U(J)	= Umsatz in Jahr J
UK(J)	= "unvorhergesehene Kosten" in Jahr J
W	Wahrheitswert

Nachdem ein Umsatz zufällig bestimmt wurde, werden die liquiden Mittel am Ende eines Jahres nach folgendem Schema ermittelt:

	Liquide Mittel des Vorjahres bzw. Liquiditätsreserve
+	Einnahmen aus Umsatzerlösen
	Ausgaben für variable Kosten (= kv * Umsatzerlöse)
-	Auszahlungen für Kreditzinsen
-	Tilgungen für Kredite
-	Privatentnahmen
-	Auszahlungen für Betriebskosten (fixe Kosten, z.B. Personal, Miete)
-	unvorhergesehene Ausgaben
=	Liquide Mittel am Ende des Jahres

Schema 4.1: Berechnung der liquiden Mittel

Die "unvorhergesehenen Ausgaben" werden als zufälliger Wert zwischen 0 und 20% der Betriebskosten des laufenden Jahres ermittelt. Sie sollen der Tatsache Rechnung tragen, daß Kostenschätzungen häufig zu niedrig liegen, weil einige Kostenarten dem Gründer unbekannt sind oder vergessen wurden, und zudem eine Reserve für unvorhergesehene Ereignisse darstellen. Alle übrigen Werte werden als vollständig bekannt vorausgesetzt.

Ergibt sich für die liquiden Mittel am Ende des Jahres ein negativer Wert, so wird der Simulationslauf abgebrochen. In diesem Fall ist das Ergebnis des Simulationslaufs eine "Pleite" innerhalb der ersten fünf Jahre. Insgesamt werden einhundert Simulationsläufe durchgeführt. Die Zahl der Simulationsläufe, bei denen eine "Pleite" auftrat, wird als Bewertungsgrundlage für die Aussage "die Liquidität_ist gesichert" herangezogen.

Tritt in mehr als 50% der Simulationsläufe eine Pleite ein, so wird die Aussage mit 0 bewertet. Tritt keine einzige Pleite ein, so ist der Wahrheitswert der Aussage 1. Dazwischen nimmt der Wahrheitswert mit steigender Zahl der Pleiten linear ab. Dies entspricht der Formel

$$W(P) = 1 - P/50$$

wobei P die Zahl der Pleiten in 100 Simulationsläufen und W (P) der Wahrheitswert der Aussage in Abhängigkeit von der Pleitenzahl ist.

Das dargestellte Simulationsmodell dient in erster Linie Demonstrationszwecken. Sein praktischer Wert für die Beurteilung der zukünftigen Liquidität eines neu zu gründenden Unternehmens kann angezweifelt werden, da es in vielen Fällen schwierig sein dürfte, die erforderlichen Einnahmen- und Ausgabengrößen auf fünf Jahre im voraus zu schätzen.

Zudem werden die Liquiditätsschwankungen innerhalb eines Jahres vernachlässigt, da es sich um ein zeitorientiertes[1] Simulationsmodell handelt. Es könnte daher eine Pleite durch Illiquidität (z.B. aufgrund eines mehrmonatigen Zahlungsverzuges eines Hauptabnehmers) auch dann eintreten, wenn alle Werte genau vorher bekannt wären und das Simulationsmodell keine Pleite errechnen würde.

Weitere Kritikpunkte könnten darauf abzielen, daß nicht alle relevanten Größen berücksichtigt werden[2] und daß die Zahl der Simulationsläufe nicht ausreicht, um statistisch gesicherte Erkenntnisse ableiten zu können. Dem letzten Einwand kann entgegengehalten werden, daß

1) Im Gegensatz zur ereignisorientierten Vorgehensweise, bei der jede einzelne Zahlung berücksichtigt würde.
2) So wird z.B. kein Forderungsausfall berücksichtigt.

eine Begrenzung der Rechenzeit des Modells im Interesse eines höheren Benutzerkomforts notwendig war und die Ungenauigkeit der Eingangsdaten ohnehin keine statistisch gesicherten Erkenntnisse zuläßt.

Trotz seiner Unzulänglichkeiten beinhaltet das dargestellte Simulationsmodell nach Meinung des Verfassers einen erheblichen Erkenntniswert. Zum einen zeigt es, daß auch numerische Werte innerhalb eines Expertensystems erfaßt und mit Hilfe komplexer mathematischer Methoden verarbeitet werden können, was für die Entwicklung betriebswirtschaftlicher Expertensysteme von großer Bedeutung ist.

Zum anderen macht das Modell deutlich, wie wichtig die Berücksichtigung einer ausreichenden Liquiditätsreserve für den Gründer ist. Ein Experiment mit einem aus einem praktischen Ratgeber entnommenen und um Umsatzhöchst- und -mindestwerte ergänzten Datensatz[1] zeigte, daß eine Erhöhung der anfänglichen Liquiditätsreserve von 0 auf DM 20.000.- eine Reduzierung der "Pleitenzahl" um etwa die Hälfte bewirkte.

Zum Schluß soll noch auf eine Erweiterungsmöglichkeit des Simulationsmodells hingewiesen werden, auf die hier aus Zeitgründen verzichtet wurde. Es wäre möglich, den Zeitpunkt der Pleite bei der Auswertung zu berücksichtigen und so ein "kritisches Jahr" zu ermitteln, in dem besonderes viele Pleiten auftraten. Ein solches Jahr könnte z.B. dadurch verursacht werden, daß viele Kredite der öffentlichen Hand die ersten beiden Jahre hindurch tilgungsfrei sind und somit die Tilgungsbelastungen im dritten Jahr sprunghaft ansteigen[2].

---

1) Vgl. Stück (85), S.91 ff.
2) Vgl. Bundesministerium für Wirtschaft (103), S.52 ff.

4.3 Kritische Würdigung des erstellten Systems

Das vorgestellte Expertensystem FOUNDER zeigt, daß Expertensysteme dazu eingesetzt werden können, ein Konzept zur Existenzgründung zu prüfen und fehlendes Wissen des Gründers aufzuspüren und zu ergänzen. Damit können sie dazu beitragen, Fehler in der Gründungsplanung zu vermeiden, und so eine erfolgreiche, langfristig tragfähige Existenz sichern helfen.

Da das Expertensystem FOUNDER in erster Linie zu Demonstrationszwecken erstellt wurde, sind nicht alle bei der Existenzgründung relevanten Sachverhalte in ausreichender Form berücksichtigt worden. Dennoch deckt die Wissensbasis das breite Spektrum betriebswirtschaftlicher Problembereiche der Existenzgründung ab. Damit wurde gleichzeitig gezeigt, daß sich Expertensysteme prinzipiell auch in anderen betriebswirtschaftlichen Problembereichen einsetzen lassen.

Das vorgestellte System wurde inklusive der theoretischen Vorarbeiten innerhalb von etwa drei Mannmonaten erstellt. Dies zeigt, daß mit verhältnismäßig wenig Aufwand ein Expertensystem-Prototyp erstellt werden konnte, an dem sich die Einsetzbarkeit der Expertensystem-Technologie im untersuchten Bereich überprüfen ließ. Voraussetzung dafür war jedoch, daß eine Expertensystem-Shell zur Verfügung stand und der Verfasser sich bereits eingehend mit der Expertensystem-Technologie befaßt hatte.

Der Umfang der Wissensbasis dürfte ausreichen, um dem Existenzgründer bei der überprüfung seines ersten groben Gründungskonzeptes zu helfen und Verbesserungsmöglichkeiten aufzuzeigen. Dennoch müßte das System für einen praktischen Einsatz, etwa als Beratungssystem in einem Amt für Wirtschaftsförderung oder einer Bank,

noch wesentlich überarbeitet, erweitert und verbessert werden. Zu diesem Zweck wäre die Hinzuziehung weiterer Experten verschiedener Fachgebiete (z.B. Gewerberecht oder Finanzierung) wünschenswert. Ebenso fehlt bisher eine umfassende Validierung der Wissensbasis anhand von praktischen Fallbeispielen.

Das Endergebnis einer intensiven Erweiterung der Wissensbasis könnte es sein, eine Fülle von "Sub-Expertensystemen", die einzelne Teilaufgaben lösen und möglicherweise mit Hilfe verschiedener Experten erstellt wurden, in einem Expertensystem zu integrieren. Solche Subsysteme könnten z.B. die Bereiche "Wahl der Rechtsform", "Standortanalyse", "Finanzierungsplan" etc. umfassen. Auch die bereits vorhandenen Teilbereiche "öffentliche Mittel" und "personenbezogene formale Voraussetzungen" könnten als derartige - wenn auch sehr einfache - Sub-Expertensysteme angesehen werden.

Das Beispiel hat jedoch auch gezeigt, daß Expertensysteme nicht unabhängig von menschlichen Beratern ein komplettes Gründungskonzept erstellen können. Die Vielzahl der im speziellen Anwendungsfall zu berücksichtigenden Faktoren würde den Rahmen des heute Realisierbaren bei weitem sprengen. Zudem ist gerade in diesem Bereich auch ein hohes Maß an Allgemeinwissen und Kreativität notwendig, etwa um die Eignung eines Standortes einzuschätzen oder eine Werbestrategie zu entwickeln. Expertensysteme können hier Gründer und Berater unterstützen, aber menschliche Problemlösungs- und Planungsfähigkeiten nicht voll ersetzen.

5 Ausblick

In dieser Arbeit wurde versucht, Probleme und Möglichkeiten einer Anwendung der Expertensystem-Technologie für betriebswirtschaftliche Problemstellungen anhand von praktischen Beispielen darzustellen. Abschließend ist zu fragen, wie sich diese Technologie in Zukunft entwickeln wird und welche Konsequenzen dies für die betriebswirtschaftliche Anwendung haben könnte. Dabei sollen keine Mutmaßungen oder Prognosen über zukünftige Marktentwicklungen aufgestellt, sondern lediglich denkbare und in Ansätzen bereits verwirklichte Entwicklungslinien aufgezeigt werden.

Zwei grundlegende Tendenzen können dabei unterschieden werden: zum einen die hard- und softwaretechnische Entwicklung, die die Leistungsfähigkeit von Expertensystemen in Zukunft verbessern und die möglichen Einsatzgebiete erweitern könnte; zum anderen die praktische Erprobung der Expertensystem-Technologie, die Erfahrungs- und Lerneffekte bringt und damit erst die Voraussetzung für eine wirtschaftlich sinnvolle Anwendung schafft.

Hinsichtlich der Hardware-Technologie zeichnen sich zwei Entwicklungen ab, die vermutlich in den nächsten Jahren zu ersten praktisch anwendbaren Ergebnissen führen werden und für die Anwendung von Expertensystemen von Bedeutung sind. Dies sind zum einen die Entwicklung leistungsfähiger Speichermedien nach dem Prinzip der Laserabtastung und zum anderen die Entwicklung von sog. Parallelrechnern[1], die die Rechengeschwindigkeit heutiger Hochleistungsrechner um ein Vielfaches übertreffen sollen.

1) von Zelewski spricht von "nebenläufigen Automatenarchitekturen" ((99), S.314 ff.)

Die Entwicklung erster sog. optischer Speichermedien, die nach dem Prinzip der CD-Schallplatte arbeiten, ist bereits abgeschlossen. Sie sind als "CD-ROM´s" auf dem Markt erhältlich[1] und bieten ein Vielfaches der Speicherkapazität konventioneller Magnetspeicherplatten. Sie können jedoch nur einmal beschrieben und danach nur noch gelesen werden, so daß sie vor allem geeignet sind, umfangreiche Datenbanken von allgemeinem Interesse (z.B. technische Normenkataloge, Wörterbücher, Lexika etc.) zur Verfügung zu stellen.

Der wesentliche Vorteil der CD-ROM´s ist es, daß sie umfangreiches Wissen in einer Form zur Verfügung stellen, in der es von Expertensystemen direkt genutzt werden kann. Allerdings ist die Aktualität der Datenbanken problematisch, da eine Aktualisierung nur durch eine Neuauflage der Datenträger erreicht werden kann. Inwieweit sich diese Technologie für betriebswirtschaftliche Expertensysteme nutzen läßt, wird davon abhängen, in welchem Maße betriebswirtschaftlich relevante Datenbanken (z.B. Marktforschungsdaten oder Daten des Statistischen Bundesamtes) auf CD-ROM´s veröffentlicht werden.

Eine Erweiterung des Laserabtast-Prinzips dahingehend, daß die Speichermedien mehrfach beschrieben werden können, ist jedoch abzusehen[2]. Dies würde einen Durchbruch in der Massenspeicher-Technologie bedeuten, der den zur Verfügung stehenden Speicherplatz vervielfachen würde. Damit wäre es auch möglich, sehr umfangreiche Wissensbasen für Expertensysteme aufzubauen oder eine Vielzahl von einzelnen Expertensystemen auf einem Speichermedium zur Verfügung zu haben, die von einem übergeordneten System mit der Lösung von Teilproblemen betraut werden.

1) Vgl. o.V. (63), S.36 f.
2) Vgl. o.V. (65), S.38 ff.

Eine weitere wichtige technische Neuerung ist die Entwicklung von sog. Parallelrechnern, die vor allem durch das japanische "FGCS-Projekt"[1] initiiert wurde. Ziel ist die Entwicklung von Computern, die Programme nicht mehr wie bisher sequentiell, sondern (teilweise) parallel abarbeiten und so ein Vielfaches der heute möglichen Rechengeschwindigkeit erreichen.

Die Parallelverarbeitung benötigt jedoch Software-Strukturen, die sich von den konventionellen, prozedural orientierten Programmiersprachen grundlegend unterscheiden. Hier können neue "deklarative" Programmiersprachen wie die Sprache PROLOG, die als Grundlage für die Softwareentwicklung des japanischen FGCS-Projektes ausgewählt wurde[2], einen entscheidenden Beitrag leisten.

Allein aus dieser Tatsache läßt sich ableiten, daß die Parallelrechner vor allem für die Entwicklung der künstlichen Intelligenz von großer Bedeutung sein werden. Es ist jedoch noch nicht abzusehen, wann die Entwicklung der Parallelrechner zu konkreten, praktisch anwendbaren Ergebnissen führt.

Betrachtet man die zu erwartende technische Entwicklung der nächsten Dekade, so lassen sich für die Anwendung der Expertensystem-Technologie folgende Schlußfolgerungen ziehen:

1) Leistungsfähigere Hardware mit größerem Speicherplatz und kürzeren Rechenzeiten wird die Entwicklung größerer und komfortablerer Expertensysteme erlauben.

---

1) FGCS für: Fifth Generation Computer Systems; vgl. Moto-Oka (57).
2) Vgl. Fuchi (24), S.116.

2) Die Software-Entwicklung wird vermutlich nicht im selben Tempo fortschreiten wie die Entwicklung der Hardware, so daß die Entwicklung neuer und besserer Methoden zur Wissensdarstellung langsam verlaufen wird. Dies führt dazu, daß der technische Fortschritt nicht voll auf die Verbesserung der Expertensystem-Technologie durchschlägt.

3) Schließlich wird der entscheidende Engpaßfaktor bei der Erstellung von Expertensystemen der Mensch sein. Zum einen zeichnet sich bereits ein Engpaß an Knowledge Engineers, die im Umgang mit der Expertensystem-Technologie geschult sind, ab[1]. Zum anderen sind vor allem die Techniken zur Gewinnung und Aufbereitung von Expertenwissen noch sehr schwach entwickelt[2]. Gerade auf diesem Gebiet wäre eine Intensivierung der Forschung, z.B. durch Psychologen oder Informationswissenschaftler, notwendig.

Die praktische Erprobung der Expertensystem-Technologie erfolgt zur Zeit in zahlreichen Pilotprojekten[3], von denen sich auch viele mit betriebswirtschaftlichen Problemstellungen beschäftigen. Der Sinn derartiger Projekte ist es,

- die Möglichkeiten und Grenzen der Anwendbarkeit der Expertensystem-Technologie zu erforschen,

- Erfahrungen bezüglich der dabei auftretenden Probleme und Schwierigkeiten zu sammeln sowie

---

1) Vgl. o.V. (64), S.14 f.
2) Vgl. von Zelewski (99), S.1165.
3) Vgl. z.B. die Aufstellung von Mertens, Allgeyer, Däsin (53).

- Problembereiche zu identifizieren, in denen Expertensysteme einen wirtschaftlichen Nutzen bringen und sie von solchen Bereichen abzugrenzen, in denen eine Anwendung zwar prinzipiell möglich, aber nicht lohnend ist.

Dabei ist nach Ansicht des Verfassers zu erwarten, daß die Ergebnisse dieser Projekte manche hochgesteckten Erwartungen enttäuschen und einige zunächst interessant erscheinende Anwendungen als unrealistisch oder unrentabel erkannt werden. Gleichzeitig werden sich jedoch auch solche Bereiche herauskristallisieren, in denen die Expertensystem-Technologie einen festen Platz im Problemlösungs-Instrumentarium der entsprechenden Fachgebiete einnehmen wird.

Die aktuelle Situation im Bereich der Expertensystem-Technologie ist in mancher Hinsicht mit der Situation des Operations Research gegen Ende der fünfziger Jahre zu vergleichen[1]. Damals wie heute wurde versucht, lohnende Anwendungsgebiete für die noch jungen Methoden zu finden. Einer anfänglichen Euphorie folgte die Erkenntnis, daß der "Stein der Weisen" doch noch nicht gefunden war und sich viele Probleme weiterhin einer Lösung widersetzten. Dennoch trugen die Methoden des Operations Research zur Bewältigung vieler bis dahin unlösbarer praktischer Probleme bei, und es wurden typische Problemklassen identifiziert, in denen einzelne Methoden besonders nutzbringend eingesetzt werden konnten.

Überträgt man die damalige Entwicklung auf die Zukunft, so ist zu erwarten, daß sich Expertensysteme in der Praxis nur vergleichsweise langsam durchsetzen werden. Gegenüber einer immer wieder euphorisch prophezeiten (und in dieser überstürzten Form kaum wünschenswerten)

1) Vgl. Fordyce, Norden, Sullivan (22), S.34 f.

"Revolution der Informationstechnologie" scheint eine vorsichtige Evolution wahrscheinlich[1]. Dies belegt auch die Tatsache, daß heute nur ein Bruchteil der entwickelten Expertensysteme tatsächlich in der Praxis eingesetzt wird[2]. Daraus sollte jedoch nicht der falsche Schluß gezogen werden, es lohne sich zum gegenwärtigen Zeitpunkt noch nicht, eigene Experimente im Bereich der Expertensysteme zu unternehmen.

Zum einen ist nämlich für die Einsetzbarkeit von Expertensystemen eine Kenntnis des im Problembereich relevanten Wissens und seiner Strukturen wichtig, die erst durch praktische Anwendung der Expertensystem-Technologie sinnvoll gewonnen werden kann. Für die spätere Konstruktion praktisch einsetzbarer Expertensysteme ist also eine frühzeitige Beschäftigung mit dieser Technologie im angestrebten Problembereich erforderlich.

Zum anderen sollten die bereits mehrfach erwähnten Seiteneffekte, die durch die Konstruktion eines Prototyp-Expertensystems entstehen, in ihrer positiven Wirkung nicht unterschätzt werden. Es scheint, als hätte die Entwicklung der künstlichen Intelligenz dazu geführt, daß erstmals in größerem Umfang erforscht wird, was menschliches Wissen eigentlich ist, obwohl die Erkenntnis der Strukturen und Zusammenhänge problemspezifischen Wissens auch ohne eine Realisierung als Computerprogramm bereits erhebliche Vorteile bringt[3].

---

1) Vgl. von Zelewski (99), S.25 ff.
2) Vgl. z.B. Mertens, Allgeyer, Däs (53), S.936.
3) Vgl. dazu die Bemerkungen von Mertens und Allgeyer betreffend die Bedeutung der expliziten Darstellung von Wissen in (52), S.706 f.

Da Wissen oder "Know How", auch wenn es sich nicht in monetären Größen quantifizieren läßt, ein wesentlicher Bestandteil der betrieblichen Leistungserstellung ist, läßt sich das "Wissen über Wissen" in seiner Bedeutung durchaus mit dem Wissen über betriebswirtschaftliche Sachverhalte und Zusammenhänge vergleichen.

Anhang A

Wissensbasis des Expertensystems

FOUNDER

```
******************
*   TITELSEITE   *
******************

                    **** FOUNDER ****
    Ein Prototyp-Expertensystem zur Beratung bei Unternehmensgründungen
         (Version 0.2 vom 18.5.87 *** (c) Karl-Ludwig von Wendt)

 Dieses Prototyp-Expertensystem soll Ihnen helfen, Ihr Konzept zur Neugrün-
 dung eines eigenen Unternehmens zu überprüfen. Es versucht, durch gezielte
 Fragen Schwachstellen in Ihrer Konzeption aufzudecken. Es ist speziell für
 solche Neugründungen gedacht, die für den Gründer den ersten Schritt in die
 Selbständigkeit darstellen.

 Bitte beantworten Sie alle Fragen in Ihrem eigenen Interesse offen und ehr-
 lich. Bedenken Sie aber auch, daß dieses System nur ein Prototyp-System ist
 und nicht alle relevanten Aspekte berücksichtigen kann. Sprechen Sie nach
 der Beratung UNBEDINGT mit einem Berater bei Ihrem zuständigen Amt für Wirt-
 schaftsförderung, der zuständigen Handwerks- oder Industrie- und Handels-
 Kammer (IHK) oder einem freien Unternehmensberater.

 Die Wissensbasis für dieses Programm wurde mit Hilfe von Herrn
 Ltd.Gew.Dir.a.D. Dr. Reinhard Schneider, Vorsitzender der Senior-Experten
 Nordwest und Mitglied des Sprechergremiums der Bundesarbeitsgemeinschaft
 der Senior-Experten, Bonn, erstellt.                  Drücken Sie jetzt [F10]


******************
*     REGELN     *
******************

Regel Nr.1 (Glaubwürdigkeit 1) :
Gründung ist sinnvoll
   wenn Gründung ist möglich --> Regel(n) 2
   und Gründung ist erfolgversprechend --> Regel(n) 5
   und Gründung ist lohnend --> Regel(n) 9

Regel Nr.2 (Glaubwürdigkeit 1) :
Gründung ist möglich
   wenn persönliche_Voraussetzungen sind erfüllt --> Regel(n) 3
   und formale_Voraussetzungen sind erfüllt --> Regel(n) 4

Regel Nr.3 (Glaubwürdigkeit 1) :
persönliche_Voraussetzungen sind erfüllt
   wenn Gründer ist körperlich_belastbar --> Regel(n) 14, 13
   und Gründer ist geistig_belastbar --> Regel(n) 17
   und Familie ist hilfsbereit --> Regel(n) 15, 16
   und Gründer hat Fachkenntnisse --> Regel(n) 19, 18
   und Gründer hat Kenntnisse_der_Betriebsführung --> Regel(n) 21, 20

Regel Nr.4 (Glaubwürdigkeit 1) :
formale_Voraussetzungen sind erfüllt
   wenn Gründer erfüllt gesetzliche_Anforderungen --> Regel(n) 24, 23, 22
   und sachliche_Voraussetzungen sind gegeben --> Fragetext
```

```
Regel Nr.5 (GlaubwÜrdigkeit 1) :
GrÜndung ist erfolgversprechend
   wenn Markt ist erfolgversprechend --> Regel(n) 6
   und Unternehmenskonzept ist erfolgversprechend --> Regel(n) 7
   und Finanzierung ist gesichert --> Regel(n) 8

Regel Nr.6 (GlaubwÜrdigkeit 1) :
Markt ist erfolgversprechend
   wenn GrÜnder hat Marktkenntnis --> Regel(n) 26, 25
   und Marktpotential ist ausreichend --> Regel(n) 29, 28, 27
   und USP ist vorhanden --> Regel(n) 33, 32, 31, 30
   und nicht Konkurrenz ist ÜbermÄchtig --> Regel(n) 36, 35, 34
   und Standort ist geeignet --> Regel(n) 38, 37
   und Beschaffung ist gesichert --> Regel(n) 39
   und Distribution ist gesichert --> Regel(n) 41, 40
   und Werbung ist ausreichend --> Regel(n) 42

Regel Nr.7 (GlaubwÜrdigkeit 1) :
Unternehmenskonzept ist erfolgversprechend
   wenn Rechtsform ist geeignet --> Regel(n) 43
   und Partner sind geeignet --> Regel(n) 44
   und Planung ist ausreichend --> Regel(n) 45
   und Know_How ist ausreichend --> Regel(n) 49
   und Alternativen sind bekannt --> Regel(n) 52
   und Personal ist geregelt --> Regel(n) 54, 53
   und Beratung ist gewährleistet --> Regel(n) 55
   und BuchfÜhrung ist gewährleistet --> Regel(n) 60, 59, 58, 57

Regel Nr.8 (GlaubwÜrdigkeit 1) :
Finanzierung ist gesichert
   wenn Sicherheiten sind ausreichend --> Regel(n) 62, 61
   und Öffentliche_Mittel sind ausgeschöpft --> Regel(n) 63, 91
   und Versicherungen sind geregelt --> Fragetext
   und die LiquiditÄt_ist gesichert --> keine Referenz!

Regel Nr.9 (GlaubwÜrdigkeit 1) :
GrÜndung ist lohnend
   wenn Motive sind ausreichend --> Regel(n) 10
   und nicht berufliche_Situation spricht gegen_GrÜndung --> Regel(n) 11, 12

Regel Nr.10 (GlaubwÜrdigkeit 1) :
Motive sind ausreichend
   wenn Leistungsbereitschaft ist vorhanden --> Fragetext
   und Verzichtbereitschaft ist vorhanden --> Fragetext
   und Verantwortungsbereitschaft ist vorhanden --> Fragetext
   und nicht Sicherheitsstreben ist hoch --> Fragetext
   und nicht Freizeitanspruch ist hoch --> Fragetext

Regel Nr.11 (GlaubwÜrdigkeit 0.7) :
berufliche_Situation spricht gegen_GrÜndung
   wenn GrÜnder ist mit_Beruf_zufrieden --> Fragetext
   und Gehalt ist sehr_hoch --> Fragetext

Regel Nr.12 (GlaubwÜrdigkeit 0.8) :
berufliche_Situation spricht gegen_GrÜndung
   wenn GrÜnder ist mit_Beruf_zufrieden --> Fragetext
   und Aufstiegschancen sind sehr_gut --> Fragetext

Regel Nr.13 (GlaubwÜrdigkeit 1) :
GrÜnder ist körperlich_belastbar
   wenn GrÜnder treibt regelmÄssig_Sport --> Fragetext
```

```
Regel Nr.14 (GlaubwÜrdigkeit 0.8) :
GrÜnder ist kÖrperlich_belastbar
   wenn nicht GrÜnder ist krank --> Fragetext
   und GrÜnder fÜhlt_sich fit --> Fragetext

Regel Nr.15 (GlaubwÜrdigkeit 1) :
Familie ist hilfsbereit
   wenn nicht GrÜnder hat Familie --> Fragetext

Regel Nr.16 (GlaubwÜrdigkeit 1) :
Familie ist hilfsbereit
   wenn GrÜnder hat Familie --> Fragetext
   und GrÜnder hat mit_Familie_gesprochen --> Fragetext
   und Familie ist einverstanden --> Fragetext

Regel Nr.17 (GlaubwÜrdigkeit 1) :
GrÜnder ist geistig_belastbar
   wenn GrÜnder kann_sich dauerhaft_konzentrieren --> Fragetext
   und GrÜnder behÄlt Überblick --> Fragetext
   und nicht GrÜnder hat Angst_vor_Stress --> Fragetext

Regel Nr.18 (GlaubwÜrdigkeit 1) :
GrÜnder hat Fachkenntnisse
   wenn GrÜnder hat abgeschlossene_Fachausbildung --> Fragetext

Regel Nr.19 (GlaubwÜrdigkeit 1) :
GrÜnder hat Fachkenntnisse
   wenn GrÜnder hat Branchen_Erfahrung --> Fragetext

Regel Nr.20 (GlaubwÜrdigkeit 1) :
GrÜnder hat Kenntnisse_der_BetriebsfÜhrung
   wenn GrÜnder hat kaufmÄnnische_Ausbildung --> Fragetext

Regel Nr.21 (GlaubwÜrdigkeit 1) :
GrÜnder hat Kenntnisse_der_BetriebsfÜhrung
   wenn nicht GrÜnder hat kaufmÄnnische_Ausbildung --> Fragetext
   und GrÜnder sagt er_habe_Kenntnisse --> Fragetext

Regel Nr.22 (GlaubwÜrdigkeit 1) :
GrÜnder erfÜllt gesetzliche_Anforderungen
   wenn Betrieb ist Handwerk --> Fragetext
   und GrÜnder ist Handwerksmeister --> Fragetext

Regel Nr.23 (GlaubwÜrdigkeit 1) :
GrÜnder erfÜllt gesetzliche_Anforderungen
   wenn Betrieb ist Handwerk --> Fragetext
   und nicht GrÜnder ist Handwerksmeister --> Fragetext
   und GrÜnder stellt einen_Meister_ein --> Fragetext

Regel Nr.24 (GlaubwÜrdigkeit 1) :
GrÜnder erfÜllt gesetzliche_Anforderungen
   wenn nicht Betrieb ist Handwerk --> Fragetext
   und Bestimmungen sind geprÜft --> Regel(n) 68

Regel Nr.25 (GlaubwÜrdigkeit 1) :
GrÜnder hat Marktkenntnis
   wenn GrÜnder hat Branchen_Erfahrung --> Fragetext
```

```
Regel Nr.26 (Glaubwürdigkeit 1) :
Gründer hat Marktkenntnis
   wenn Gründer kennt führende_Konkurrenten --> Fragetext
   und Gründer kennt Marktrisiken --> Fragetext

Regel Nr.27 (Glaubwürdigkeit 1) :
Marktpotential ist ausreichend
   wenn Bedarf ist vorhanden --> Fragetext
   und Produkt ist bekannt --> Fragetext
   und nicht Markt ist gesättigt --> Fragetext

Regel Nr.28 (Glaubwürdigkeit 1) :
Marktpotential ist ausreichend
   wenn Marktanalyse ist durchgeführt --> Fragetext
   und Marktanalyse ist positiv --> Fragetext

Regel Nr.29 (Glaubwürdigkeit 1) :
Marktpotential ist ausreichend
   wenn Produkt ist innovativ --> Fragetext
   und Bedarf ist vorhanden --> Fragetext

Regel Nr.30 (Glaubwürdigkeit 1) :
USP ist vorhanden
   wenn Produkt hat besseres_P_L_Verhältnis --> Fragetext

Regel Nr.31 (Glaubwürdigkeit 1) :
USP ist vorhanden
   wenn Produkt ist innovativ --> Fragetext

Regel Nr.32 (Glaubwürdigkeit 1) :
USP ist vorhanden
   wenn Produkt ist Marktlücke --> Fragetext

Regel Nr.33 (Glaubwürdigkeit 1) :
USP ist vorhanden
   wenn Qualität ist ausserordentlich --> Fragetext

Regel Nr.34 (Glaubwürdigkeit 1) :
Konkurrenz ist übermächtig
   wenn kleinere_Geschäfte mussten aufgeben --> Fragetext

Regel Nr.35 (Glaubwürdigkeit 1) :
Konkurrenz ist übermächtig
   wenn Grossunternehmen bietet selbe_Leistung_an --> Fragetext

Regel Nr.36 (Glaubwürdigkeit 1) :
Konkurrenz ist übermächtig
   wenn es herrscht Verdrängungswettbewerb --> Fragetext

Regel Nr.37 (Glaubwürdigkeit 1) :
Standort ist geeignet
   wenn Standortanalyse ist durchgeführt --> Fragetext
   und Standortanalyse ist positiv --> Fragetext

Regel Nr.38 (Glaubwürdigkeit 1) :
Standort ist geeignet
   wenn nicht Standortanalyse ist durchgeführt --> Fragetext
   und Verkehrsanbindung ist gut --> Fragetext
   und Kundschaft ist vorhanden --> Fragetext
   und nicht Nachbarschaft ist störend --> Fragetext
   und Einzugsgebiet ist gross_genug --> Fragetext
   und Konkurrenz ist gering --> Fragetext
   und nicht Standortqualität wird_sich verschlechtern --> Fragetext
   und Kosten sind angemessen --> Fragetext
   und Standortalternativen sind bekannt --> Fragetext
```

```
Regel Nr.39 (GlaubwÜrdigkeit 1) :
Beschaffung ist gesichert
   wenn Lieferanten sind bekannt --> Fragetext
   und nicht GrÜnder ist abhÄngig_von_Lieferanten --> Fragetext

Regel Nr.40 (GlaubwÜrdigkeit 1) :
Distribution ist gesichert
   wenn nicht Betrieb ist Industrie --> Regel(n) 66

Regel Nr.41 (GlaubwÜrdigkeit 1) :
Distribution ist gesichert
   wenn Betrieb ist Industrie --> Regel(n) 66
   und Absatzkanal ist bekannt --> Fragetext
   und Absatzkanal ist offen --> Fragetext

Regel Nr.42 (GlaubwÜrdigkeit 1) :
Werbung ist ausreichend
   wenn nicht Werbung wird unterschÄtzt --> Regel(n) 67
   und Werbestrategie ist ausgearbeitet --> Fragetext

Regel Nr.43 (GlaubwÜrdigkeit 1) :
Rechtsform ist geeignet
   wenn Rechtsform ist bekannt --> Fragetext
   und Rechtsform ist mit_Berater_abgesprochen --> Fragetext

Regel Nr.44 (GlaubwÜrdigkeit 1) :
Partner sind geeignet
   wenn Partner sind vorhanden --> Fragetext
   und VertrauensverhÄltnis besteht zwischen_den_Partnern --> Fragetext
   und Partner erfÜllen persönliche_Voraussetzungen --> Fragetext
   und Kompetenzen sind geklÄrt --> Fragetext
   und Leistungsverhältnis ist ausgewogen --> Fragetext

Regel Nr.45 (GlaubwÜrdigkeit 1) :
Planung ist ausreichend
   wenn Finanzplan ist vorhanden --> Fragetext
   und LiquiditÄtsplan ist vorhanden --> Fragetext
   und Marketingkonzept ist vorhanden --> Fragetext
   und nicht Bedeutung_der_Planung wird unterschÄtzt --> Regel(n) 46
   und Planung ist detailliert --> Regel(n) 48
   und nicht Planung ist blauÄugig --> Regel(n) 47

Regel Nr.46 (GlaubwÜrdigkeit 1) :
Bedeutung_der_Planung wird unterschÄtzt
   wenn GrÜnder stimmt Aussage1_zu --> Fragetext

Regel Nr.47 (GlaubwÜrdigkeit 1) :
Planung ist blauÄugig
   wenn GrÜnder kennt Umsatz_in_3_Jahren_genau --> Fragetext

Regel Nr.48 (GlaubwÜrdigkeit 1) :
Planung ist detailliert
   wenn GrÜnder kennt Betriebsausgaben_in_Jahr_1_genau --> Fragetext

Regel Nr.49 (GlaubwÜrdigkeit 1) :
Know_How ist ausreichend
   wenn GrÜnder ist auf_neuestem_Stand --> Fragetext
   und GrÜnder verfolgt Fachpresse --> Fragetext
```

```
Regel Nr.52 (GlaubwÜrdigkeit 1) :
Alternativen sind bekannt
   wenn Übernahme ist geprüft --> Fragetext
   und Franchising ist geprüft --> Fragetext
   und Kooperation ist geprüft --> Fragetext

Regel Nr.53 (GlaubwÜrdigkeit 1) :
Personal ist geregelt
   wenn nicht Personal wird benötigt --> Fragetext

Regel Nr.54 (GlaubwÜrdigkeit 1) :
Personal ist geregelt
   wenn Personal wird benötigt --> Fragetext
   und Arbeitsrecht ist bekannt --> Fragetext
   und Personal ist zu_bekommen --> Fragetext

Regel Nr.55 (GlaubwÜrdigkeit 1) :
Beratung ist gewährleistet
   wenn GrÜnder kennt Rechtsanwalt --> Fragetext
   und GrÜnder kennt Steuerberater --> Fragetext
   und nicht Beratung wird unterschätzt --> Regel(n) 56

Regel Nr.56 (GlaubwÜrdigkeit 1) :
Beratung wird unterschätzt
   wenn GrÜnder stimmt Aussage2_zu --> Fragetext

Regel Nr.57 (GlaubwÜrdigkeit 1) :
BuchfÜhrung ist gewährleistet
   wenn GrÜnder beherrscht BuchfÜhrung --> Fragetext
   und GrÜnder kennt BuchfÜhrungspflicht --> Fragetext

Regel Nr.58 (GlaubwÜrdigkeit 1) :
BuchfÜhrung ist gewährleistet
   wenn GrÜnder beschäftigt Buchhalter --> Fragetext

Regel Nr.59 (GlaubwÜrdigkeit 1) :
BuchfÜhrung ist gewährleistet
   wenn GrÜnder kennt mobilen_BuchfÜhrungshelfer --> Fragetext

Regel Nr.60 (GlaubwÜrdigkeit 1) :
BuchfÜhrung ist gewährleistet
   wenn GrÜnder will BuchfÜhrungskurs_besuchen --> Fragetext

Regel Nr.61 (GlaubwÜrdigkeit 1) :
Sicherheiten sind ausreichend
   wenn Sicherheiten entsprechen Kapitalbedarf --> Fragetext

Regel Nr.62 (GlaubwÜrdigkeit 1) :
Sicherheiten sind ausreichend
   wenn GrÜnder hat mit_Bank_gesprochen --> Fragetext
   und Kredit ist zugesagt --> Fragetext

Regel Nr.63 (GlaubwÜrdigkeit 1) :
öffentliche_Mittel sind ausgeschöpft
   wenn GrÜnder wÜnscht Mittel_Beratung --> Fragetext
   und Eigenkapitalhilfe ist geprÜft --> Regel(n) 93, 92
   und ERP_Darlehen ist geprÜft --> Regel(n) 95, 94
   und ergänzungsprogramme sind geprÜft --> Regel(n) 100, 99, 98
   und BÜrgschaften sind geprÜft --> Regel(n) 103, 102, 101
   und Beratungsförderung ist geprÜft --> Regel(n) 104, 105
   und Hinweis auf LÄnderprogramme --> Regel(n) 106
```

```
Regel Nr.64 (GlaubwÜrdigkeit 1) :
Finanzplanung ist ausreichend
   wenn Durststrecke ist berÜcksichtigt --> Fragetext
   und LiquiditÄtsreserve ist berÜcksichtigt --> Fragetext
   und Zahlungsverzug ist berÜcksichtigt --> Fragetext

Regel Nr.65 (GlaubwÜrdigkeit 1) :
Bilanzrelation ist gut
   wenn langfristige_Mittel decken Anlagevermögen --> Fragetext

Regel Nr.66 (GlaubwÜrdigkeit 1) :
Betrieb ist Industrie
   wenn nicht Betrieb ist Handwerk --> Fragetext
   und Frage_Betrieb ist Industrie --> Fragetext

Regel Nr.67 (GlaubwÜrdigkeit 1) :
Werbung wird unterschÄtzt
   wenn GrÜnder stimmt Aussage3_zu --> Fragetext

Regel Nr.68 (GlaubwÜrdigkeit 1) :
Bestimmungen sind geprÜft
   wenn Reisegewerbekarte ist geprÜft --> Regel(n) 70, 69
   und Sachkundenachweis ist geprÜft --> Regel(n) 74, 73
   und besondere_Genehmigung ist geprÜft --> Regel(n) 77, 78

Regel Nr.69 (GlaubwÜrdigkeit 1) :
Reisegewerbekarte ist geprÜft
   wenn GrÜnder benÖtigt Reisegewerbekarte --> Regel(n) 88, 86
   und GrÜnder hat Reisegewerbekarte --> Fragetext

Regel Nr.70 (GlaubwÜrdigkeit 1) :
Reisegewerbekarte ist geprÜft
   wenn nicht GrÜnder benÖtigt Reisegewerbekarte --> Regel(n) 88, 86

Regel Nr.71 (GlaubwÜrdigkeit 1) :
GrÜnder benÖtigt Sachkundenachweis
   wenn Betrieb ist Einzelhandel --> Regel(n) 83
   und Warenart gehört_zu Gruppe_3 --> Fragetext

Regel Nr.72 (GlaubwÜrdigkeit 1) :
GrÜnder benÖtigt Sachkundenachweis
   wenn Betrieb ist Gastwirtschaft --> Regel(n) 89

Regel Nr.73 (GlaubwÜrdigkeit 1) :
Sachkundenachweis ist geprÜft
   wenn GrÜnder benÖtigt Sachkundenachweis --> Regel(n) 72, 71, 80
   und GrÜnder hat Sachkundenachweis --> Fragetext

Regel Nr.74 (GlaubwÜrdigkeit 1) :
Sachkundenachweis ist geprÜft
   wenn nicht GrÜnder benÖtigt Sachkundenachweis --> Regel(n) 72, 71, 80

Regel Nr.75 (GlaubwÜrdigkeit 1) :
GrÜnder benÖtigt besondere_Genehmigung
   wenn Betrieb ist freiberuflich --> Regel(n) 87
   und nicht Beruf ist genehmigungsfrei --> Regel(n) 76

Regel Nr.76 (GlaubwÜrdigkeit 1) :
Beruf ist genehmigungsfrei
   wenn GrÜnder hat_sich Über_Genehmigung_informiert --> Fragetext
   und nicht Genehmigung ist erforderlich --> Fragetext
```

```
Regel Nr.77 (GlaubwÜrdigkeit 1) :
besondere_Genehmigung ist geprüft
   wenn Gründer benötigt besondere_Genehmigung --> Regel(n) 90, 75, 85
   und Gründer hat besondere_Genehmigung --> Fragetext

Regel Nr.78 (GlaubwÜrdigkeit 1) :
besondere_Genehmigung ist geprüft
   wenn nicht Gründer benötigt besondere_Genehmigung --> Regel(n) 90, 75, 85

Regel Nr.79 (GlaubwÜrdigkeit 1) :
Gründer benötigt besondere_Erlaubnis
   wenn Betrieb ist Grosshandel --> Regel(n) 84
   und Warenart gehört zu_Gruppe_1 --> Fragetext

Regel Nr.80 (GlaubwÜrdigkeit 1) :
Gründer benötigt Sachkundenachweis
   wenn Betrieb ist Einzelhandel --> Regel(n) 83
   und Warenart gehört zu_Gruppe_1 --> Fragetext

Regel Nr.81 (GlaubwÜrdigkeit 1) :
Betrieb ist Handel
   wenn nicht Betrieb ist Handwerk --> Fragetext
   und nicht Betrieb ist Industrie --> Regel(n) 66
   und Frage_Betrieb ist Handel --> Fragetext

Regel Nr.82 (GlaubwÜrdigkeit 1) :
Betrieb ist Dienstleistung
   wenn nicht Betrieb ist Handwerk --> Fragetext
   und nicht Betrieb ist Handel --> Regel(n) 81
   und nicht Betrieb ist Industrie --> Regel(n) 66
   und nicht Frage_Betrieb ist Landwirtschaft --> Fragetext

Regel Nr.83 (GlaubwÜrdigkeit 1) :
Betrieb ist Einzelhandel
   wenn Betrieb ist Handel --> Regel(n) 81
   und Frage_Betrieb ist Einzelhandel --> Fragetext

Regel Nr.84 (GlaubwÜrdigkeit 1) :
Betrieb ist Grosshandel
   wenn Betrieb ist Handel --> Regel(n) 81
   und nicht Betrieb ist Einzelhandel --> Regel(n) 83

Regel Nr.85 (GlaubwÜrdigkeit 1) :
Gründer benötigt besondere_Genehmigung
   wenn Betrieb ist Dienstleistung --> Regel(n) 82
   und nicht Betrieb ist freiberuflich --> Regel(n) 87
   und Branche gehört zu_Gruppe_2 --> Fragetext

Regel Nr.86 (GlaubwÜrdigkeit 1) :
Gründer benötigt Reisegewerbekarte
   wenn Betrieb ist Handel --> Regel(n) 81
   und Betrieb ist mobil --> Fragetext

Regel Nr.87 (GlaubwÜrdigkeit 1) :
Betrieb ist freiberuflich
   wenn Betrieb ist Dienstleistung --> Regel(n) 82
   und nicht Branche gehört zu_Gruppe_2 --> Fragetext
   und Frage_Betrieb ist freiberuflich --> Fragetext
```

```
Regel Nr.88 (GlaubwÜrdigkeit 1) :
GrÜnder benötigt Reisegewerbekarte
   wenn Betrieb ist Dienstleistung --> Regel(n) 82
   und nicht Betrieb ist freiberuflich --> Regel(n) 87
   und nicht Branche gehört zu_Gruppe_2 --> Fragetext
   und Betrieb ist Jahrmarkt_Bude --> Fragetext

Regel Nr.89 (GlaubwÜrdigkeit 1) :
Betrieb ist Gastwirtschaft
   wenn Betrieb ist Dienstleistung --> Regel(n) 82
   und nicht Branche gehört zu_Gruppe_2 --> Fragetext
   und nicht Betrieb ist freiberuflich --> Regel(n) 87
   und Frage_Betrieb ist Gastwirtschaft --> Fragetext

Regel Nr.90 (GlaubwÜrdigkeit 1) :
GrÜnder benötigt besondere_Genehmigung
   wenn Betrieb ist Gastwirtschaft --> Regel(n) 89

Regel Nr.91 (GlaubwÜrdigkeit 1) :
öffentliche_Mittel sind ausgeschöpft
   wenn nicht GrÜnder wÜnscht Mittel_Beratung --> Fragetext

Regel Nr.92 (GlaubwÜrdigkeit 1) :
Eigenkapitalhilfe ist geprÜft
   wenn Eigenkapitalhilfe kommt in_Frage --> Regel(n) 107

Regel Nr.93 (GlaubwÜrdigkeit 1) :
Eigenkapitalhilfe ist geprÜft.

Regel Nr.94 (GlaubwÜrdigkeit 1) :
ERP_Darlehen ist geprÜft
   wenn ERP_Darlehen kommt in_Frage --> Regel(n) 108

Regel Nr.95 (GlaubwÜrdigkeit 1) :
ERP_Darlehen ist geprÜft.

Regel Nr.96 (GlaubwÜrdigkeit 1) :
ergänzungsprogramm1 kommt in_Frage
   wenn nicht Vorhaben ist bereits_begonnen --> Fragetext

Regel Nr.97 (GlaubwÜrdigkeit 1) :
ergänzungsprogramm2 kommt in_Frage
   wenn GrÜnder ist Spätaussiedler --> Fragetext

Regel Nr.98 (GlaubwÜrdigkeit 1) :
ergänzungsprogramme sind geprÜft
   wenn ergänzungsprogramm1 kommt in_Frage --> Regel(n) 96

Regel Nr.99 (GlaubwÜrdigkeit 1) :
ergänzungsprogramme sind geprÜft
   wenn ergänzungsprogramm2 kommt in_Frage --> Regel(n) 97

Regel Nr.100 (GlaubwÜrdigkeit 1) :
ergänzungsprogramme sind geprÜft.

Regel Nr.101 (GlaubwÜrdigkeit 1) :
BÜrgschaften sind geprÜft
   wenn BÜrgschaft_fÜr_Freiberufler kommt in_Frage --> Regel(n) 109
```

```
Regel Nr.102 (Glaubwürdigkeit 1) :
Bürgschaften sind geprüft
   wenn Kreditgarantiegemeinschaft kommt in_Frage --> Regel(n) 110

Regel Nr.103 (Glaubwürdigkeit 1) :
Bürgschaften sind geprüft.

Regel Nr.104 (Glaubwürdigkeit 1) :
Beratungsförderung ist geprüft
   wenn Beratungshilfe kommt in_Frage --> Regel(n) 111

Regel Nr.105 (Glaubwürdigkeit 1) :
Beratungsförderung ist geprüft.

Regel Nr.106 (Glaubwürdigkeit 1) :
Hinweis auf Länderprogramme.

Regel Nr.107 (Glaubwürdigkeit 1) :
Eigenkapitalhilfe kommt in_Frage
   wenn Investitionssumme ist_grösser_als 40000 --> Fragetext
   und Eigenmittel sind_mehr_als zwölf_Prozent --> Fragetext
   und nicht Eigenmittel sind ausreichend --> Fragetext
   und Gründer hat Fachkenntnisse --> Regel(n) 19, 18
   und Gründer hat Kenntnisse_der_Betriebsführung --> Regel(n) 21, 20
   und nicht Vorhaben ist bereits_begonnen --> Fragetext

Regel Nr.108 (Glaubwürdigkeit 1) :
ERP_Darlehen kommt in_Frage
   wenn nicht Vorhaben ist bereits_begonnen --> Fragetext
   und Sicherheiten sind ausreichend --> Regel(n) 62, 61

Regel Nr.109 (Glaubwürdigkeit 1) :
Bürgschaft_für_Freiberufler kommt in_Frage
   wenn Betrieb ist freiberuflich --> Regel(n) 87
   und nicht Sicherheiten sind ausreichend --> Regel(n) 62, 61

Regel Nr.110 (Glaubwürdigkeit 0.5) :
Kreditgarantiegemeinschaft kommt in_Frage
   wenn nicht Sicherheiten sind ausreichend --> Regel(n) 62, 61

Regel Nr.111 (Glaubwürdigkeit 1) :
Beratungshilfe kommt in_Frage
   wenn Gründer hatte Beratung --> Fragetext
   und Beratung ist bezahlt --> Fragetext
   und Beratung gibt Aufschluss_über_Gründungserfolg --> Fragetext
```

```
******************
*   FRAGETEXTE   *
******************
```

Fragetext für die Aussage 'Gründer kennt Buchführungspflicht ':
Wissen Sie, in welchem Umfang Sie verpflichtet sind, ihre Geschäftsvorgänge aufzuzeichnen?

Fragetext für die Aussage 'Liquiditätsreserve ist berücksichtigt ':
Haben Sie in Ihrer Planung eine ausreichende Liquiditätsreserve vorgesehen, um unerwartete Ausgaben oder Umsatzeinbrüche abfangen zu können?

Fragetext für die Aussage 'Produkt hat besseres_P_L_Verhältnis ':
Glauben Sie, daß Sie ihren Kunden ein deutlich besseres Preis-Leistungs-Verhältnis als die Konkurrenz bieten können?

Fragetext für die Aussage 'Beratung gibt Aufschluss_Über_Gründungserfolg ':
Gibt das Ergebnis der Beratung Aufschluß darüber, ob die Gründung für Sie lohnend und erfolgversprechend ist?

Fragetext für die Aussage 'Beratung ist bezahlt ':
Ist diese Beratung bereits von Ihnen bezahlt worden?

Fragetext für die Aussage 'Gründer hatte Beratung ':
Haben Sie sich von einem professionellen Unternehmensberater entgeltlich über Ihr Gründungsvorhaben beraten lassen?

Fragetext für die Aussage 'Eigenmittel sind ausreichend ':
Sind Ihre Eigenmittel ausreichend, um eine nachhaltige und tragfähige Existenzgründung zu ermöglichen?

Fragetext für die Aussage 'Eigenmittel sind_mehr_als zwölf_Prozent ':
Betragen Ihre Eigenmittel mehr als zwölf Prozent der gesamten Investitionssumme?

Fragetext für die Aussage 'Investitionssumme ist_grösser_als 40000 ':
Ist die Gesamtsumme der erforderlichen Investitionen grösser als DM 40.000,- ?

Fragetext für die Aussage 'Gründer ist Spätaussiedler ':
Sind Sie Spätaussiedler oder Spätberechtigter, der nach dem 31.12.1970 in die Bundesrepublik oder Westberlin gekommen ist?

Fragetext für die Aussage 'Vorhaben ist bereits_begonnen ':
Haben Sie mit Ihrem Gründungsvorhaben bereits begonnen, d.h. haben Sie z.B. bereits Waren oder Maschinen bestellt?

Fragetext für die Aussage 'Gründer wünscht Mittel_Beratung ':
Wünschen Sie eine Beratung, welche öffentlichen Mittel für die Finanzierung Ihres Projektes in Frage kommen?

Fragetext für die Aussage 'Branche gehört zu_Gruppe_2 ':
Ist Ihr zukünftiger Betrieb einer der folgenden Branchen zuzurechnen: Gütertransport, Personenbeförderung, Immobilienvermittlung, Kredit- oder Versicherungsvermittlung, Versteigerung, Bewachungsgewerbe, Automaten-Aufstellung bzw. Spielhalle, Fahrschule?

Fragetext für die Aussage 'Frage_Betrieb ist Gastwirtschaft ':
Wollen Sie eine Gastwirtschaft, einen Imbißbetrieb oder ähnliches eröffnen?

Fragetext für die Aussage 'Warenart gehört_zu Gruppe_5 ':
Wollen Sie ein Einzelhandelsgeschäft im Bereich Reformwaren oder eine Drogerie eröffnen?

Fragetext für die Aussage 'Betrieb ist Jahrmarkt_Bude ':
Möchten Sie einen Stand auf einem Jahrmarkt oder ähnlichen Veranstaltungen mit variablem Standort einrichten?

Fragetext für die Aussage 'Genehmigung ist erforderlich ':
Ist eine Genehmigung erforderlich?

Fragetext für die Aussage 'Gründer hat_sich über_Genehmigung_informiert ':
Haben Sie sich darüber informiert, ob für Ihren Beruf eine behördliche Genehmigungspflicht besteht?

Fragetext für die Aussage 'Frage_Betrieb ist freiberuflich ':
Wollen Sie sich freiberuflich selbständig machen (z.B. als Arzt, Rechtsanwalt, Übersetzer)?

Fragetext für die Aussage 'Gründer hat Reisegewerbekarte ':
Haben Sie eine Reisegewerbekarte beantragt oder erhalten?

Fragetext für die Aussage 'Gründer hat Sachkundenachweis ':
Können Sie den erforderlichen Nachweis der Sachkunde (z.B. durch eine abgeschlossene Berufsausbildung oder eine Prüfung bei der IHK) erbringen?

Fragetext für die Aussage 'Gründer hat besondere_Genehmigung ':
Haben Sie die erforderliche Genehmigung beantragt oder erhalten?

Fragetext für die Aussage 'Betrieb ist mobil ':
Beabsichtigen Sie, Ihre Kunden direkt zu besuchen und Ihre Geschäfte an Ort und Stelle abzuschließen?

Fragetext für die Aussage 'Frage_Betrieb ist Einzelhandel ':
Wollen Sie ein Einzelhandelsgeschäft eröffnen?

Fragetext für die Aussage 'Frage_Betrieb ist Landwirtschaft ':
Wollen Sie sich im Bereich der Land- oder Forstwirtschaft selbständig machen?

Fragetext für die Aussage 'Frage_Betrieb ist Handel ':
Wollen Sie einen Handelsbetrieb eröffnen?

Fragetext für die Aussage 'Warenart gehört zu_Gruppe_1 ':
Umfasst Ihr zukünftiges Sortiment auch Milch, Waffen oder unedle Metalle (z.B. Schrott)?

Fragetext für die Aussage 'Qualität ist ausserordentlich ':
Können Sie eine ausserordentliche und schwer nachzuahmende Qualität anbieten?

Fragetext für die Aussage 'Produkt ist Marktlücke ':
Stoßen Sie mit Ihrem Angebot in eine Marktlücke, die bisher noch nicht abgedeckt wurde?

Fragetext für die Aussage 'Produkt hat besseres ':
Hat Ihr Produkt/Ihre Leistung im Vergleich zur Konkurrenz ein deutlich besseres Preis-Leistungs-Verhältnis?

Fragetext für die Aussage 'Produkt ist innovativ ':
Ist Ihr Produkt bzw. Leistung innovativ, d.h. in dieser Form und auf diesem Markt noch nicht dagewesen?

Fragetext für die Aussage 'Marktanalyse ist positiv ':
Ist das Ergebnis der Analyse positiv?

Fragetext für die Aussage 'Marktanalyse ist durchgeführt ':
Haben Sie von einem unabhängigen Gutachter, z.B. bei der IHK, eine Analyse Ihrer Marktchancen machen lassen?

Fragetext für die Aussage 'Markt ist gesättigt ':
Ist der Markt für Ihre Leistung bereits ausreichend abgedeckt?

Fragetext für die Aussage 'Produkt ist bekannt ':
Sind Ihre potentiellen Kunden mit der besonderen Art Ihrer Leistung vertraut und wissen Sie diese zu schätzen?

Fragetext für die Aussage 'Bedarf ist vorhanden ':
Gibt es für Ihr Produkt/Ihre Leistung eigentlich einen ausreichenden Bedarf?

Fragetext für die Aussage 'Gründer kennt Marktrisiken ':
Wissen Sie, auf welche Faktoren es in Ihrem Markt besonders ankommt, wenn man Erfolg haben will?

Fragetext für die Aussage 'Gründer kennt führende_Konkurrenten ':
Kennen Sie die wichtigsten Konkurrenten in Ihrem Markt?

Fragetext für die Aussage 'sachliche_Voraussetzungen sind gegeben ':
Sind Sie sicher, daß Sie alle eventuellen gesetzlichen Bestimmungen und Auflagen für Ihre Betriebsstätte kennen und erfüllen können?

Fragetext für die Aussage 'Gründer stellt einen_Meister_ein ':
Beabsichtigen Sie, in Ihrem Betrieb einen Handwerksmeister einzustellen?

Fragetext für die Aussage 'Gründer ist Handwerksmeister ':
Haben Sie die für Ihren Betrieb erforderliche handwerkliche Meisterprüfung abgelegt?

Fragetext für die Aussage 'Gründer sagt er_habe_Kenntnisse ':
Haben Sie die nötigen Kenntnisse, um einen Betrieb zu führen, Preise zu kalkulieren und Entscheidungen sicher und schnell zu treffen?

Fragetext für die Aussage 'Gründer hat kaufmännische_Ausbildung ':
Haben Sie eine kaufmännische Ausbildung abgeschlossen?

Fragetext für die Aussage 'Gründer hat Branchen_Erfahrung ':
Haben Sie bereits mehrere Jahre in der Branche Ihres neuen Unternehmens gearbeitet?

Fragetext für die Aussage 'Gründer hat abgeschlossene_Fachausbildung ':
Haben Sie eine abgeschlossene Berufsausbildung in dem Bereich, in dem Sie ihr Unternehmen gründen wollen?

Fragetext für die Aussage 'Gründer kennt Betriebsausgaben_in_Jahr_1_genau ':
Können Sie heute bereits sagen, welche Ausgaben für Miete, Heizung und Personal Sie im achten Monat nach der Gründung vermutlich bezahlen müssen?

Fragetext für die Aussage 'Gründer kennt Umsatz_in_3_Jahren_genau ':
Wissen Sie schon heute, welchen Umsatz Sie im September des dritten Jahres nach der Gründung machen werden?

Fragetext für die Aussage 'Gründer stimmt Aussage1_zu ':
Glauben Sie, daß die Aussage
'Planung kostet viel Zeit und bringt wenig. Letztlich kann man ja doch nicht wissen, wie's kommt.'
zutrifft?

Fragetext für die Aussage 'Marketingkonzept ist vorhanden ':
Haben Sie sich ein Konzept gemacht, wie Sie sich am Markt gegen die Konkurrenz durchsetzen und einen ausreichenden Kundenstamm aufbauen können?

Fragetext für die Aussage 'Liquiditätsplan ist vorhanden ':
Haben Sie für mindestens die ersten zwölf Monate einen Liquiditätsplan erstellt, der gewährleistet, daß Sie immer genug Geld in der Kasse haben, um Ihren Verbindlichkeiten nachkommen zu können?

Fragetext für die Aussage 'Finanzplan ist vorhanden ':
Haben Sie einen ausführlichen Finanzierungsplan erstellt, der auch Raum für unvorhergesehene Änderungen hat?

Fragetext für die Aussage 'Leistungsverhältnis ist ausgewogen ':
Ist gewährleistet, daß jeder Partner in dem Maße Gewinn aus dem Unternehmen zieht, wie er sich persönlich und finanziell dafür einsetzt?

Fragetext für die Aussage 'Kompetenzen sind geklärt ':
Haben Sie alle Pflichten, Kompetenzen und Rechte in gemeinsamem Einvernehmen genau geklärt und vertraglich festgehalten?

Fragetext für die Aussage 'Partner erfüllen persönliche_Voraussetzungen ':
Bringen alle an der Gründung Beteiligten die erforderlichen geistigen, körperlichen und materiellen Voraussetzungen mit?

Fragetext für die Aussage 'Vertrauensverhältnis besteht zwischen_den_Partnern ':
Kennen Sie Ihre(n) Partner gut? Besteht ein solides Vertrauensverhältnis zwischen Ihnen?

Fragetext für die Aussage 'Partner sind vorhanden ':
Beabsichtigen Sie, die Gründung gemeinsam mit einem oder mehreren Partnern durchzuführen?

Fragetext für die Aussage 'Rechtsform ist mit_Berater_abgesprochen ':
Haben Sie die Wahl der Rechtsform mit einem Steuerberater besprochen?

Fragetext für die Aussage 'Rechtsform ist bekannt ':
Wissen Sie, welche Rechtsform Ihr Unternehmen annehmen soll?

Fragetext für die Aussage 'Gründer stimmt Aussage3_zu ':
Würden Sie der folgenden Aussage
'Werbung ist herausgeworfenes Geld. Wenn man gut ist, spricht sich das auch so schnell herum.'
zustimmen?

Fragetext für die Aussage 'Werbestrategie ist ausgearbeitet ':
Haben Sie bereits eine Strategie ausgearbeitet, wie Sie Ihr Produkt oder Ihre Leistung den Kunden bekannt machen wollen?

Fragetext für die Aussage 'Absatzkanal ist offen ':
Sind Sie sicher, daß dieser Weg für Sie offensteht und Ihnen keine Hindernisse in den Weg gelegt werden?

Fragetext für die Aussage 'Absatzkanal ist bekannt ':
Wissen Sie, auf welchem Wege Ihre Erzeugnisse zum Endverbraucher gelangen sollen?

Fragetext für die Aussage 'Frage_Betrieb ist Industrie ':
Wollen Sie einen Produktions- bzw. Industriebetrieb gründen?

Fragetext für die Aussage 'Gründer ist abhängig_von_Lieferanten ':
Besteht die Gefahr, daß Sie sich durch Inanspruchnahme von Sonderkonditionen von einem Lieferanten abhängig machen?

Fragetext für die Aussage 'Lieferanten sind bekannt ':
Kennen Sie die wichtigsten Lieferanten in Ihrem Bereich?

Fragetext für die Aussage 'Standortalternativen sind bekannt ':
Haben Sie mindestens drei mögliche Standorte auf Ihre unterschiedliche Eignung geprüft, bevor Sie sich entschieden haben?

Fragetext für die Aussage 'Kosten sind angemessen ':
Entsprechen die Kosten für Miete oder Pacht der Standortqualität?

Fragetext für die Aussage 'Standortqualität wird_sich verschlechtern ':
Besteht die Gefahr, daß sich die Situation an Ihrem Standort (z.B. aufgrund Städtebaulicher Maßnahmen) in absehbarer Zeit wesentlich verschlechtern könnte?

Fragetext für die Aussage 'Konkurrenz ist gering ':
Bietet die Konkurrenzlage an Ihrem Standort noch ausreichend "Luft" für einen weiteren Betrieb der gleichen Branche?

Fragetext für die Aussage 'Einzugsgebiet ist gross_genug ':
Bietet Ihr Standort ein ausreichend grosses Einzugsgebiet (das ist das Gebiet, in dem es für die Kunden mindestens ebenso bequem ist, bei Ihnen zu kaufen oder zu bestellen wie bei der Konkurrenz)?

Fragetext für die Aussage 'Nachbarschaft ist störend ':
Gibt es in der Nähe Ihres Standortes Einrichtungen oder Geschäfte, die sich nachteilig auf Ihr Image oder den Kundenstrom auswirken könnten?

Fragetext für die Aussage 'Kundschaft ist vorhanden ':
Gibt es an Ihrem Standort ausreichend Kunden, bzw. läßt die Geschäftslage ausreichend Laufkundschaft erwarten?

Fragetext für die Aussage 'Verkehrsanbindung ist gut ':
Ist die Verkehrsanbindung (Transportmöglichkeiten für Lieferanten, Kunden-Parkplätze usw.) an Ihrem Standort ausreichend?

Fragetext für die Aussage 'Standortanalyse ist positiv ':
Kam das Gutachten zu dem Schluß, daß der Standort für Ihr Vorhaben geeignet ist?

Fragetext für die Aussage 'Standortanalyse ist durchgeführt ':
Haben Sie von einem unabhängigen Gutachter eine Standortanalyse durchführen lassen?

Fragetext für die Aussage 'es herrscht Verdrängungswettbewerb ':
Ist es auf Ihrem Markt in letzter Zeit zu aggressivem Wettbewerb und Preiskampf gekommen?

Fragetext für die Aussage 'Grossunternehmen bietet selbe_Leistung_an ':
Wird Ihre Leistung oder Ihr Produkt in Ihrem Absatzbereich in ähnlicher Form auch von einem Grossunternehmen angeboten, oder besteht die Gefahr, daß ein solches Unternehmen in Kürze (z.B. durch Filialeröffnung) in Ihren Absatzbereich eindringt?

Fragetext für die Aussage 'kleinere_Geschäfte mussten aufgeben ':
Ist es in Ihrem Markt in letzter Zeit häufiger zu Schließungen von kleineren Betrieben gekommen?

Fragetext für die Aussage 'langfristige_Mittel decken Anlagevermögen ':
Ist Ihr Anlagevermögen (Maschinen, Einrichtung, Fahrzeuge etc.) vollständig durch langfristige Kredite (Laufzeit mindestens so hoch wie Nutzungsdauer der entsprechenden Gegenstände) oder Eigenkapital finanziert?

Fragetext für die Aussage 'Zahlungsverzug ist berücksichtigt ':
Haben Sie berücksichtigt, daß Ihre Kunden evt. Ihre Leistungen erst mit einer Verzögerung von mehreren Monaten bezahlen könnten?

Fragetext für die Aussage 'Durststrecke ist berücksichtigt ':
Haben Sie bei Ihrer Finanzplanung auch die sog. Durststrecke (die ersten 6-12 Monate) berücksichtigt, in der Ihre Umsätze noch niedrig, Ihre Kosten aber besonders hoch sind?

Fragetext für die Aussage 'Kredit ist zugesagt ':
Haben Sie bei Ihrer Bank eine Zusage für Kredite in ausreichender Höhe erhalten?

Fragetext für die Aussage 'Gründer hat mit_Bank_gesprochen ':
Haben Sie bereits mit einem Kreditberater Ihrer Bank über Ihr Vorhaben gesprochen?

Fragetext für die Aussage 'Sicherheiten entsprechen Kapitalbedarf ':
Können Sie Ihren Bedarf an Startkapital durch bankübliche Sicherheiten (z.B. Hypothek, Bürgschaften, Übereignung von Maschinen etc.) ausreichend absichern?

Fragetext für die Aussage 'Gründer will Buchführungskurs_besuchen ':
Beabsichtigen Sie, sich durch Besuch eines Kursus (z.B. bei der IHK) selbst die nötigen Buchführungskenntnisse anzueignen?

Fragetext für die Aussage 'Gründer kennt mobilen_Buchführungshelfer ':
Kennen Sie jemanden, der Ihnen (entgeltlich oder unentgeltlich) bei Ihrer Buchführung helfen kann?

Fragetext für die Aussage 'Gründer beschäftigt Buchhalter ':
Beabsichtigen Sie, eine Arbeitskraft für die Buchführung einzustellen?

Fragetext für die Aussage 'Gründer beherrscht Buchführung ':
Beherrschen Sie die Techniken der modernen Buchführung?

Fragetext für die Aussage 'Gründer stimmt Aussage2_zu ':
Stimmen Sie der folgenden Aussage
'Beratung ist zu Beginn viel zu teuer; außerdem kenne ich mich selbst gut genug aus!'
zu?

Fragetext für die Aussage 'Gründer kennt Steuerberater ':
Kennen Sie einen qualifizierten Steuerberater?

Fragetext für die Aussage 'Gründer kennt Rechtsanwalt ':
Kennen Sie einen guten Rechtsanwalt?

Fragetext für die Aussage 'Personal ist zu_bekommen ':
Sind Sie sicher, daß sie das benötigte Personal in der erforderlichen Qualifikation bekommen können?

Fragetext für die Aussage 'Arbeitsrecht ist bekannt ':
Haben Sie sich gründlich über die für Sie bedeutsamen Vorschriften des Arbeitsrechtes informiert?

Fragetext für die Aussage 'Personal wird benötigt ':
Müssen Sie für Ihr Unternehmen Personal einstellen?

Fragetext für die Aussage 'Kooperation ist geprüft ':
Haben Sie die Möglichkeit geprüft, sich mit anderen Gründern zusammenzuschliessen?

Fragetext für die Aussage 'Übernahme ist geprüft ':
Haben Sie überlegt, ob nicht auch eine Übernahme eines bereits bestehenden Betriebes für Sie in Betracht kommt?

Fragetext für die Aussage 'Franchising ist geprüft ':
Haben Sie geprüft, ob es auch die Möglichkeit gibt, Ihr Unternehmen als Franchise-Betrieb zu eröffnen?

Fragetext für die Aussage 'Übernahme ist geplant ':
Beabsichtigen Sie, ein bereits bestehendes Unternehmen zu übernehmen?

Fragetext für die Aussage 'Betrieb ist Handwerk ':
Wollen Sie einen Handwerksbetrieb gründen?

Fragetext für die Aussage 'Gründer verfolgt Fachpresse ':
Lesen Sie regelmässig Fachzeitschriften und/oder Verbandsmitteilungen in ihrem Fachgebiet?

Fragetext für die Aussage 'Gründer ist auf_neuestem_Stand ':
Kennen Sie die neueste technische Entwicklung bzw. die modernsten Hilfsmittel in Ihrem Fachgebiet?

Fragetext für die Aussage 'Gründer hat Angst_vor_Stress ':
Haben Sie Angst davor, übermässig belastet oder in Anspruch genommen zu werden?

Fragetext für die Aussage 'Gründer behält Überblick ':
Behalten Sie auch in schwierigen Situationen stets Ruhe und den nötigen Überblick?

Fragetext für die Aussage 'Gründer kann_sich dauerhaft_konzentrieren ':
Können Sie sich über einen längeren Zeitraum auf eine Aufgabe konzentrieren, ohne sich von Nebensächlichkeiten ablenken zu lassen?

Fragetext für die Aussage 'Familie ist einverstanden ':
Unterstützt Ihre Familie Ihr Vorhaben ohne Vorbehalte?

Fragetext für die Aussage 'Gründer hat mit_Familie_gesprochen ':
Haben Sie mit Ihrer Familie über Ihr Vorhaben gesprochen?

Fragetext für die Aussage 'Gründer hat Familie ':
Haben Sie Familie?

Fragetext für die Aussage 'Gründer fühlt_sich fit ':
Fühlen Sie sich fit genug, um über einen längeren Zeitraum eine 60-Stunden-Woche durchzuhalten?

Fragetext für die Aussage 'Gründer ist krank ':
Leiden Sie an chronischen Erkrankungen (wie z.B. Herzleiden), die Ihre Leistungsfähigkeit und Belastbarkeit beeinträchtigen?

Fragetext für die Aussage 'Gründer treibt regelmässig_Sport ':
Treiben Sie regelmässig (mindestens 1 mal in der Woche) Sport?

Fragetext für die Aussage 'Aufstiegschancen sind sehr_gut ':
Sind die Aufstiegschancen für Sie in Ihrem bisherigen Arbeitsverhältnis überdurchschnittlich gut?

Fragetext für die Aussage 'Gehalt ist sehr_hoch ':
Verdienen Sie bisher, gemessen an Ihrer Ausbildung und Qualifikation, überdurchschnittlich viel?

Fragetext für die Aussage 'Gründer ist mit_Beruf_zufrieden ':
Sind Sie mit Ihrer bisherigen beruflichen Tätigkeit zufrieden?

Fragetext für die Aussage 'Freizeitanspruch ist hoch ':
Legen Sie besonderen Wert auf langen Urlaub, freie Wochenenden und einen rechtzeitigen Feierabend?

Fragetext für die Aussage 'Sicherheitsstreben ist hoch ':
Sind Ihnen ein gesichertes Einkommen und geregelte Freizeit wichtiger als finanzieller Erfolg?

Fragetext für die Aussage 'Verantwortungsbereitschaft ist vorhanden ':
Sind Sie bereit, Verantwortung zu tragen und auch einmal für die Fehler anderer geradezustehen?

Fragetext für die Aussage 'Verzichtbereitschaft ist vorhanden ':
Sind Sie bereit, Ihre Ansprüche und Ihren Lebensstandard für eine Weile wesentlich zu reduzieren?

Fragetext für die Aussage 'Leistungsbereitschaft ist vorhanden ':
Sind Sie bereit, sich mit voller Kraft für Ihr Unternehmen einzusetzen und auch einmal auf ein Wochenende oder den Urlaub zu verzichten?

Fragetext für die Aussage 'Versicherungen sind geregelt ':
Haben Sie sich auch Gedanken über die notwendigen Versicherungen (Feuer-, Haftpflicht-, Krankenversicherung usw.) gemacht?

```
******************
*  HINWEISTEXTE  *
******************
```

Hinweistext für die Aussage 'Beschaffung ist gesichert '
(Wahrheitswert zwischen 0 und 0.5):

Der günstige Einkauf ist ein oft unterschätzter Faktor des wirtschaftlichen Erfolgs. Es ist wichtig, daß Sie unterschiedliche Lieferanten in Ihrem Markt kennen und gute Beziehungen zu Ihnen unterhalten, um Sonderkonditionen in Anspruch zu nehmen und flexibel das jeweils günstigste Angebot auswählen zu können. Achten Sie vor allem darauf, nicht durch langfristige Liefervertrage in eine Abhängigkeit zu geraten und damit Ihre Anpassungsfähigkeit an den Markt auf's Spiel zu setzen.

Hinweistext für die Aussage 'Kreditgarantiegemeinschaft kommt in_Frage '
(Wahrheitswert zwischen 0.3 und 1):

Wenn Ihre eigenen Sicherheiten nicht ausreichen, um Ihr Kreditvolumen abzusichern, können Sie sich an eine Kreditgarantiegemeinschaft Ihres Wirtschaftszweiges wenden. Diese übernimmt bis zu 80% der Deckungssumme und ermöglicht Ihnen so den Zugang zu Krediten, die Sie selbst nicht ausreichend absichern können. Bitte fragen Sie in Ihrer Handwerkskammer, der IHK, Ihrer Hausbank oder einem Wirtschaftsverband nach Einzelheiten.

Hinweistext für die Aussage 'Know_How ist ausreichend '
(Wahrheitswert zwischen 0 und 0.5):

Vorsicht: In einer Zeit sich rasch verändernder Märkte und rapiden technischen Fortschritts ist es besonders wichtig, stets "auf dem Laufenden" zu sein. Das gilt durchaus nicht nur für technologieorientierte Gründungen! Wenn Sie die neuesten Trends des Marktes nicht erkennen oder mit veralteten Methoden oder Geräten arbeiten, kann es Ihnen passieren, daß Sie am Markt vorbeiwirtschaften oder durch zu hohe Kosten von der Konkurrenz an die Wand gedrängt werden.

Hinweistext für die Aussage 'Buchführung ist gewährleistet '
(Wahrheitswert zwischen 0 und 0.5):

Für eine erfolgreiche Geschäftsführung ist die Buchführung ein unverzichtbares Mittel. Dies betrifft nicht nur die gesetzliche Pflicht zur Buchführung sondern auch die Tatsache, daß die Aufzeichnung Ihrer Geschäftsvorgänge für Sie ein äußerst wertvolles Informationsinstrument ist. Wenn Sie allein den Anforderungen der Buchführung nicht gewachsen sind, können Sie Ihren Steuerberater oder einen sog. "mobilen Buchführungshelfer" hinzuziehen, um stets den Überblick zu behalten.

Hinweistext für die Aussage 'Beratung ist gewährleistet '
(Wahrheitswert zwischen 0 und 0.5):

Für eine erfolgreiche Gründung ist es besonders wichtig, daß Sie jederzeit jemanden haben, den Sie in Schwierigkeiten um Rat fragen können. Verlassen Sie sich dabei nicht nur auf den Rat wohlmeinender Freunde und Verwandten, sondern ziehen Sie einen erfahrenen Fachmann hinzu! Sie sollten auf jeden Fall einen Rechtsanwalt und Steuerberater kennen, zu denen Sie Vertrauen haben. Ebenfalls sehr nützlich ist es, sich in bestimmten Fragen an einen Unternehmensberater zu wenden.

Hinweistext für die Aussage 'Personal ist geregelt '
(Wahrheitswert zwischen 0 und 0.5):

Das Personal ist in vielen Bereichen, vor allem im Bereich der Dienstleistungen, ein wichtiger und kritischer Faktor. Gerade in der Anfangsphase werden Sie Schwierigkeiten haben, ausreichend qualifiziertes und motiviertes Personal zu bekommen. Zudem kommt eine Vielzahl von Bestimmungen und zusätzlichen Kosten auf Sie zu, die Sie vorher genau kennen sollten! Einige dieser Schwierigkeiten können Sie umgehen, wenn Ihnen zu Beginn Familienmitglieder bei der Betriebsführung helfen.

Hinweistext für die Aussage 'Alternativen sind bekannt '
(Wahrheitswert zwischen 0 und 0.5):

Sie sollten sich noch einmal ausführlich informieren, ob nicht zu der von Ihnen gewählten Betriebsform noch Alternativen bestehen. Gerade das Franchising z.B. bietet Jungunternehmern eine Chance, ein eigenes Unternehmen zu gründen und dabei das Know How eines Großunternehmens zu nutzen. Zudem gibt es bei Ihrer IHK eine Existenzgründer-Börse, in der regelmäßig Angebote für Übernahmen, Partnerschaften oder Beteiligungen aufgeführt werden.

Hinweistext für die Aussage 'Planung ist ausreichend '
(Wahrheitswert zwischen 0 und 0.5):

Sehr viele Existenzgründer unterschätzen die Bedeutung der Planung gerade in der Gründungsphase. Sie meinen, vor lauter praktischen Problemen keine Zeit zum Planen zu haben. Diese Fehleinschätzung kann verhängnisvoll sein! Eine präzise, realistische und vollständige Planung ist für eine erfolgreiche Gründung unverzichtbar. Lassen Sie sich am besten bei Ihrer Bank, Ihrer zuständigen Handwerkskammer oder IHK beraten, was ein Gründungsplan alles enthalten muß und wie Sie dabei am besten vorgehen!

Hinweistext für die Aussage 'Partner sind geeignet '
(Wahrheitswert zwischen 0 und 0.5):

Vorsicht: Zahlreiche Praxisfälle belegen, daß die Wahl des richtigen Partners und vor allem ein zweckmäßiger schriftlicher Vertrag zwischen Ihnen von überragender Bedeutung für eine Geschäftsgründung ist. Überlassen Sie nichts dem Zufall oder dem guten Willen! Es wäre nicht das erste Mal, daß gute Freunde sich über eine relativ belanglose Angelegenheit zerstreiten und schließlich erbittert bekämpfen, was nahezu zwangsläufig zu einem Scheitern der Gründung führen muß.

Hinweistext für die Aussage 'Rechtsform ist geeignet '
(Wahrheitswert zwischen 0 und 0.5):

Die Wahl der Rechtsform Ihres Unternehmens ist eine wichtige und grundsätzliche Entscheidung, die Sie auf jeden Fall mit einem Steuerberater oder Rechtsanwalt besprechen sollten. Bedenken Sie bei der Vorüberlegung, daß die Haftungsbeschränkung bei einer GmbH Sie nicht vor der Inanspruchnahme aus Bankkrediten schützt und daß Sie als persönlich haftender Gesellschafter bzw. Einzelkaufmann bei Ihrer Bank ein höheres Ansehen genießen!

Hinweistext für die Aussage 'Werbung ist ausreichend '
(Wahrheitswert zwischen 0 und 0.5):

Vorsicht: Sie unterschätzen vielleicht die Bedeutung der Werbung für Ihr Produkt oder Ihre Dienstleistung. Ein altes Sprichwort sagt: "Wer nicht wirbt, der stirbt!". Es ist etwas Wahres daran: wie sollen Ihre Kunden zu Ihnen kommen, wenn Sie von Ihrer Leistung oder Ihrem Produkt nichts wissen? Überlegen Sie, ob nicht vielleicht kleine, aber regelmässig geschaltete Anzeigen in der Tagespresse oder Fachzeitschriften Sie einem grösseren Kundenkreis bekannt machen können!

Hinweistext für die Aussage 'Distribution ist gesichert '
(Wahrheitswert zwischen 0 und 0.5):

Offenbar herrscht bei Ihnen noch Unklarheit, wie Sie Ihre Produkte an den Endverbraucher bringen sollen. Unterschätzen Sie nicht die Schwierigkeiten, die Sie als Neuling am Markt haben, in das Sortiment des Facheinzel- oder Großhandels aufgenommen zu werden. Viele Fachgeschäfte sind in Einkaufsringen oder -gemeinschaften organisiert. Sprechen Sie einmal mit Vertretern dieser Gemeinschaften! Überlegen Sie auch, ob eine Vertriebskooperation mit anderen Kleinherstellern in Frage kommt!

Hinweistext für die Aussage 'Standort ist geeignet '
(Wahrheitswert zwischen 0 und 0.5):

Sie sollten Ihren Standort noch einmal gründlich überprüfen und sich überlegen, ob es allen Unbequemlichkeiten zum Trotz nicht doch besser wäre, auch einen Standort in einer anderen Stadt in Erwägung zu ziehen. Wenn Sie bisher noch nicht wenigstens drei verschiedene Standortalternativen ins Auge gefaßt haben, sollten Sie dies jetzt unbedingt nachholen! Bedenken Sie, daß Sie eine Standortwahl später praktisch nicht mehr rückgängig machen können!

Hinweistext für die Aussage 'Konkurrenz ist übermächtig '
(Wahrheitswert zwischen 0.5 und 1):

Achtung: Sie sind im Begriff, sich mit übermächtigen Gegnern einzulassen! Bedenken Sie, daß ein Großkonzern jederzeit in einen Markt eindringen kann, den er bisher vernachlässigt hat, sobald dieser Markt attraktiv erscheint. Sie haben dann kaum eine Chance, sich gegen diese Bedrohung zu wehren. Versuchen Sie, die Vorteile eines Kleinbetriebes (größere Flexibilität, Kundennähe etc.) auszunutzen und wählen Sie eine Marktnische, die für ein Großunternehmen nicht attraktiv genug ist!

Hinweistext für die Aussage 'USP ist vorhanden '
(Wahrheitswert zwischen 0 und 0.5):

Ihnen fehlt offenbar "das gewisse Etwas", das Ihren künftigen Betrieb von der Konkurrenz abhebt. Überlegen Sie sich einmal, was Ihre Kunden bewegen könnte, nicht mehr wie bisher bei der Konkurrenz, sondern bei Ihnen zu kaufen bzw. zu bestellen. Versuchen Sie, sich einen besonderen Vorteil zu verschaffen (z.B. außergewöhnliche Qualität, besonderer Service, Tag- und Nachtdienst etc.), den Ihre Konkurrenz nicht hat!

Hinweistext für die Aussage 'Marktpotential ist ausreichend '
(Wahrheitswert zwischen 0 und 0.5):

Möglicherweise ist das Marktpotential in Ihrem zukünftigen Absatzbereich nicht ausreichend! Das bedeutet für Sie, daß Sie nicht genügend Umsatz erwirtschaften können, um Ihre Kosten zu decken und einen ausreichenden Gewinn zu machen und daß Sie praktisch keine Wachstumschancen haben. Sie sollten sich überlegen, ob Sie nicht ein attraktiveres Marktsegment (z.B. einen anderen Standort) auswählen sollten.

Hinweistext für die Aussage 'Gründer hat Marktkenntnis '
(Wahrheitswert zwischen 0 und 0.5):

Vorsicht: Möglicherweise mangelt es Ihnen an einem tiefgehenden Einblick in die Zusammenhänge und Gegebenheiten Ihres zukünftigen Marktes. Der Markt entscheidet über das Schicksal Ihres Unternehmens und damit über Ihre zukünftige Existenz. Wenn Sie seine Spielregeln nicht genau kennen, laufen Sie Gefahr, von den Konkurrenten verdrängt zu werden. Informieren Sie sich genau, bevor Sie eine Existenzgründung wagen! Bedenken Sie, daß die Probleme einer Branche von außen sehr leicht unterschätzt werden!

Hinweistext für die Aussage 'Beratungshilfe kommt in_Frage '
(Wahrheitswert zwischen 0.7 und 1):

Sie können für bereits bezahlte Beratungen einen Zuschuß bis zu 60% der Beratungskosten, jedoch bei Existenzgründungsberatungen höchstens DM 2.500,-, erhalten. Einen Antrag müssen Sie bei Ihrer zuständigen Leitstelle stellen. Fragen Sie bei Ihrer zuständigen Kammer nach der Adresse. Ebenso erhalten Sie einen Zuschuß für Beratungen durch das Rationalisierungs-Kuratorium der Deutschen Wirtschaft.

Hinweistext für die Aussage 'Bürgschaft_für_Freiberufler kommt in_Frage '
(Wahrheitswert zwischen 0.7 und 1):

Für Sie kommt eine Bürgschaft der Deutschen Ausgleichsbank in Frage. Damit können Sie maximal 80% der Kredite von Kreditinstituten absichern, sofern Ihre eigenen Sicherheiten nicht ausreichen. Sprechen Sie bitte mit Ihrem Kreditinstitut über diese Möglichkeit.

Hinweistext für die Aussage 'Hinweis auf Länderprogramme '
(Wahrheitswert zwischen 0 und 1):

Neben den Bundesmitteln stellen auch die Länder und Gemeinden eigene Programme zur Förderung von Existenzgründungen und Betriebsansiedlungen zur Verfügung, insbesondere auch Steuervergünstigungen. Bitte erkundigen Sie sich danach bei Ihrem zuständigen Amt für Wirtschaftsförderung.

Hinweistext für die Aussage 'Ergänzungsprogramm2 kommt in_Frage '
(Wahrheitswert zwischen 0.7 und 1):

Für Sie kommt das Ergänzungsprogramm II der Deutschen Ausgleichsbank in Frage. Das Darlehen hat einen Höchstbetrag von DM 200.000,-, einen Zins- von 5,25% und wird zu 100% ausgezahlt. Die Laufzeit beträgt 12 Jahre, da von die ersten beiden tilgungsfrei. Die Absicherung erfolgt zu banküblichen Konditionen, da Ihre Hausbank das Kreditrisiko trägt. Auf Antrag kann eine 50%ige Entlastungszusage erteilt werden (0,5% Provision p.a.).

Hinweistext für die Aussage 'Ergänzungsprogramm1 kommt in_Frage '
(Wahrheitswert zwischen 0.7 und 1):

Für Sie kommt das Ergänzungsprogramm I der Deutschen Ausgleichsbank in Frage. Sie können damit einen Betrag in Höhe von maximal 50% Ihrer Investitionssumme zu einem Zinssatz von 6% p.a. und einem Auszahlungsbetrag von 97% erhalten. Die Haftung wird von Ihrer Hausbank getragen, die daher von Ihnen eine ausreichende Absicherung fordert. Die Laufzeit beträgt in der Regel 10 Jahre, davon die ersten beiden tilgungsfrei.

Hinweistext für die Aussage 'ERP_Darlehen kommt in_Frage
(Wahrheitswert zwischen 0.7 und 1):

Sie können bei Ihrem Kreditinstitut einen Antrag auf ein Existenzgründungsdarlehen des ERP-Programms der Deutschen Ausgleichsbank stellen. Die Haftung für das Darlehen wird von Ihrer Hausbank übernommen, die daher von Ihnen eine ausreichende Absicherung fordern wird. Der Zinssatz beträgt 5,5% (im Zonenrandgebiet 4,5%), die Laufzeit maximal 10 Jahre (bei Bauvorhaben 15 Jahre). Die ersten beiden Jahre sind tilgungsfrei.

Hinweistext für die Aussage 'Eigenkapitalhilfe kommt in_Frage '
(Wahrheitswert zwischen 0.7 und 1):

Sie können einen Antrag auf Eigenkapitalhilfe der Deutschen Ausgleichsbank bei Ihrem Kreditinstitut stellen. Die Eigenkapitalhilfe braucht nicht durch Sicherheiten abgedeckt zu werden und haftet wie Eigenkapital. Sie können damit Ihre Eigenmittel auf maximal 40% der Investitionssumme, jedoch höchstens um DM 300.000,- (in Berlin 350.000,-), aufstocken. Die Konditionen sind äußerst günstig (näheres bei Ihrem Kreditinstitut).

Hinweistext für die Aussage 'Gründer benötigt Reisegewerbekarte '
(Wahrheitswert zwischen 0.7 und 1):

Für den Betrieb Ihres Gewerbes benötigen Sie eine Reisegewerbekarte, die Sie bei Ihrem Gewerbeaufsichtsamt beantragen müssen.

Hinweistext für die Aussage 'Gründer benötigt besondere_Genehmigung '
(Wahrheitswert zwischen 0.7 und 1):

Für den Betrieb Ihres Gewerbes ist eine besondere Genehmigung erforderlich, die Sie bei Ihrem zuständigen Gewerbeaufsichtsamt beantragen müssen. Dort teilt man Ihnen auch mit, welche weiteren Beschränkungen oder Voraussetzungen ggf. bestehen.

Hinweistext für die Aussage 'Gründer benötigt Sachkundenachweis '
(Wahrheitswert zwischen 0.7 und 1):

Für den Betrieb Ihres zukünftigen Gewerbes benötigen Sie den Nachweis der erforderlichen Sachkunde. Dazu müssen Sie eine abgeschlossene Berufsausbildung in Ihrem Bereich haben oder eine Prüfung vor der Industrie- und Handelskammer ablegen.

Hinweistext für die Aussage 'Gründung ist lohnend '
(Wahrheitswert zwischen 0.30001 und 0.7):

Sind Sie sicher, daß Sie alle Unannehmlichkeiten einer Gründung in Kauf nehmen und die Sicherheit und geregelte Arbeitszeit Ihrer bisherigen Beschäftigung aufgeben wollen? Bedenken Sie vor einer Entscheidung auch, daß Sie ein Mehrfaches Ihres bisherigen Gehaltes an Gewinn machen müssen, um Ihren Lebensstandard halten zu können.

Hinweistext für die Aussage 'Gründung ist lohnend '
(Wahrheitswert zwischen 0 und 0.3):

Achtung: es gibt gravierende Gründe, die gegen eine selbständige Gründung sprechen, selbst wenn diese eine gute Erfolgschance hätte. Überlegen Sie, ob Sie die Sicherheits- und Bequemlichkeitsnachteile einer selbständigen Existenz wirklich in Kauf nehmen wollen. Bedenken Sie auch, daß Sie auf Dauer ein Mehrfaches Ihres jetzigen Gehaltes an Gewinn machen müssen, um Ihren Lebensstandard halten zu können!

Hinweistext für die Aussage 'die Liquidität_ist gesichert
(Wahrheitswert zwischen 0.30001 und 0.7):

Vorsicht: Obwohl Sie eine realistische Überlebenschance haben, besteht die Gefahr einer Pleite in den ersten fünf Jahren. Überlegen Sie, am besten zusammen mit einem Fachmann Ihrer Bank, wie Sie ihre Liquiditätsreserve erhöhen können, um mit Umsatzschwankungen und unvorhergesehenen Kosten besser fertig werden zu können!

Hinweistext für die Aussage 'die Liquidität_ist gesichert '
(Wahrheitswert zwischen 0 und 0.3):

Achtung: bei Ihrem bisherigen Gründungskonzept besteht die Gefahr der Pleite durch Illiquidität innerhalb der ersten fünf Jahre nach der Gründung! Sie sollten in jedem Fall überlegen, wie Sie Ihre Liquiditätsreserve erhöhen können, um Gefahren durch Umsatzeinbrüche oder unvorhergesehene Kosten abfangen zu können. Sprechen Sie darüber mit einem Fachmann Ihrer Bank!

Hinweistext für die Aussage 'Finanzierung ist gesichert '
(Wahrheitswert zwischen 0.30001 und 0.7):

Bei der Finanzierung Ihres Unternehmens sind immer noch Schwachstellen und Unwägbarkeiten vorhanden. Nehmen Sie diese nicht auf die leichte Schulter! Sprechen Sie mit einem Fachmann Ihrer Bank darüber, wie Sie Ihr Finanzierungskonzept "wasserdicht" machen können!

Hinweistext für die Aussage 'Finanzierung ist gesichert '
(Wahrheitswert zwischen 0 und 0.3):

Achtung: Ihre Finanzierung ist nicht gesichert! Ein sehr großer Teil der Pleiten bei Existenzgründern wird durch eine unsolide Finanzierung verursacht, die unvorhergesehene Kosten oder Umsatzeinbußen nicht verkraften kann. Sprechen Sie mit einem Fachmann bei Ihrer Bank; er kann Ihnen sagen, ob eine Chance besteht, Ihr Finanzierungskonzept tragfähig zu machen!

Hinweistext für die Aussage 'Unternehmenskonzept ist erfolgversprechend '
(Wahrheitswert zwischen 0.30001 und 0.7):

Ihr Unternehmenskonzept weist noch Schwachstellen auf. Sprechen Sie das Konzept am besten noch einmal mit einem erfahrenen Berater durch und überlegen Sie, wo Sie Ihre Planung verbessern und vervollständigen können. Unterschätzen Sie nicht die Bedeutung einer sorgfältigen Gründungsplanung und -vorbereitung!

Hinweistext für die Aussage 'Unternehmenskonzept ist erfolgversprechend '
(Wahrheitswert zwischen 0 und 0.3):

Achtung: Ihr Unternehmenskonzept weist noch erhebliche Mängel auf. Prüfen Sie, wie Sie Ihre Planung und die Gestaltung Ihres Betriebes verbessern können, und ziehen Sie den Rat eines erfahrenen Fachmannes hinzu. Unterschätzen Sie auf keinen Fall die Bedeutung einer sorgfältigen Planung und Vorbereitung! Von ihr hängt Ihr Schicksal als selbständiger Unternehmer ab!

Hinweistext für die Aussage 'Markt ist erfolgversprechend '
(Wahrheitswert zwischen 0.300001 und 0.7):

Der Markt für Ihr Produkt oder Ihre Dientsleistung bietet Ihnen Chancen, aber auch erhebliche Risiken. Seien Sie bei der Einschätzung Ihrer Möglichkeiten selbstkritisch und vorsichtig. Die besten Kenntnisse und die neuesten Techniken nützen Ihnen nichts, wenn der Markt sie nicht akzeptiert. Sprechen Sie mit einem erfahrenen Berater über Ihre Marktchancen und die besonderen Gefahren, denen Sie ausgesetzt sind!

Hinweistext für die Aussage 'Markt ist erfolgversprechend '
(Wahrheitswert zwischen 0 und 0.3):

Achtung: der Markt, auf den Sie sich mit Ihrem Unternehmen begeben wollen, birgt erhebliche Risiken für Sie! Unterschätzen Sie nicht die Widerstände, die der Markt neugegründeten Unternehmen entgegenbringt, vor allem wenn der Gründer auf dem Markt unerfahren ist. Wenn Sie trotzdem zur Gründung entschlossen sind, lassen Sie am besten von einen unabhängigen Gutachter eine Analyse Ihrer Marktchancen erstellen!

Hinweistext für die Aussage 'formale_Voraussetzungen sind erfüllt '
(Wahrheitswert zwischen 0 und 0.7):

Vorsicht: es besteht die Gefahr, daß bei Ihnen nicht die gesetzlichen Voraussetzungen für die Gründung erfüllt sind. Sprechen Sie am besten mit einem Rechtsanwalt, der Sie über die einschlägigen Vorschriften und Ihre Möglichkeiten, Sie zu erfüllen, informiert.

Hinweistext für die Aussage 'persönliche_Voraussetzungen sind erfüllt '
(Wahrheitswert zwischen 0.30000001 und 0.7):

Sie bringen noch nicht alle notwendigen Voraussetzungen für eine erfolgreiche Unternehmensgründung mit. Überlegen Sie, wie Sie Ihr persönliches Umfeld oder Ihren Kenntnisstand verändern können, um erfolgreich sein zu können. Erwägen Sie auch die Hinzunahme eines Partners, der Sie bei für Sie schwierigen Aufgaben unterstützen und entlasten kann.

Hinweistext für die Aussage 'persönliche_Voraussetzungen sind erfüllt '
(Wahrheitswert zwischen 0 und 0.3):

Ihnen fehlen wichtige Voraussetzungen, die für das erfolgreiche Führen eines Unternehmens unumgänglich sind. Überlegen Sie, ob sich eine selbständige Existenz wirklich für Sie lohnt. Prüfen Sie auch, ob Sie z.B. durch Fortbildung die notwendigen Voraussetzungen für die Gründung schaffen können, oder ob nicht ein Partner die Aufgaben übernehmen kann, denen Sie vielleicht nicht gewachsen sind.

Hinweistext für die Aussage 'Gründung ist sinnvoll '
(Wahrheitswert zwischen 0 und 0.29999999):

Ihr Gründungskonzept weist noch gravierende Mängel auf! Berücksichtigen Sie die Hinweise, die ich Ihnen bereits gegeben habe, und überlegen Sie, am besten zusammen mit einem erfahrenen Berater, wie Sie die aufgetretenen Probleme überwinden können. Fragen Sie sich noch einmal, ob Ihr Konzept vollständig und ausgereift ist und ob sich eine Gründung mit all ihren Risiken wirklich für Sie lohnt!

Hinweistext für die Aussage 'Gründung ist sinnvoll '
(Wahrheitswert zwischen 0.3 und 0.69999999):

Ihr Gründungskonzept ist noch nicht frei von Schwächen. Ich habe Ihnen bereits Hinweise auf die Schwachstellen gegeben. Diese Hinweise können Sie als Anregung verstehen, Ihr Konzept zu überarbeiten und zu vervollständigen. Sprechen Sie mit einem erfahrenen Berater über die aufgetretenen Schwachstellen; er kann Ihnen vielleicht einen Weg zeigen, wie Sie die Probleme überwinden können.

Hinweistext für die Aussage 'Gründung ist sinnvoll '
(Wahrheitswert zwischen 0.7 und 1):

Herzlichen Glückwunsch! Wenn Sie alle Fragen ehrlich und realistisch beantwortet haben, weist Ihr Gründungsvorhaben keine gravierenden Schwächen oder Mängel auf. Bedenken Sie aber, daß dieses Programm nicht alle relevanten Aspekte berücksichtigen kann und sprechen Sie auf jeden Fall noch einmal mit einem erfahrenen Berater.
Ich wünsche Ihnen viel Erfolg mit Ihrem neuen Betrieb!

Anhang B

Prolog-Listing:

Die Expertensystem-Shell MOSES

```
/*************************************************************/
/*                                                           */
/*                          MOSES                            */
/*                                                           */
/*                                                           */
/*          Modulare Shell fUr Experten-Systeme              */
/*                      Version 1.0                          */
/*                                                           */
/*                        (c) 1987                           */
/*                by Karl-Ludwig von Wendt                   */
/*                                                           */
/*************************************************************/
 project "MOSES"
 code = 4800
 trail = 500

global domains

   namelist = reference name*
   name  = reference string
   konsequenz = reference dann (name, name, name)
   kons = reference konsequenz*
   bedingung  = reference  wenn (name, name, name);
                     wenn_nicht (name, name, name)
   bedingungen = reference bedingung*
   confidence = reference real
   cons = reference confidence*
   number = reference integer
   numbers = reference number*


  database
   context (name, namelist)
   rule (number, konsequenz, bedingungen, confidence)
   fact (number, namelist, confidence)
   rules (number)
   facts (number)
   question (name, name, name, string)
   conclusion (name, name, name, confidence, confidence, string)
   was_con (name, name, name, confidence)
   track (numbers)
   how_trace (symbol)
   found (symbol)
   work_rule (number)
   work_fact (number)
   header (string)
   used (kons)
   mincon (confidence)
   stop
   was_answer
   ziel (name)

/*  INTERN DEFINIERTE RELATIONEN */

  global predicates
   internal (name) - (i)
   is_internal (name) - (o)
   ist_wahr (name, name, name, confidence) - (i, i, i, o)

  predicates

/*  INFERENZMASCHINE */

   wahr (name, name, name, confidence)
   aux_wahr (name, name, name, confidence)
   erfuellt (bedingungen, confidence)
   falsch (name, name, name, confidence)
   max (cons, confidence)
   min (cons, confidence)
   set_found
```

```
    reset_found
    getmincon (confidence)
    setmincon (confidence)


/*  ERKLÄRUNGS-KOMPONENTE */

    update_track (number)
    explain_why (numbers)
    cut_track
    trace_on
    trace_off
    trace_fact (name, name, name, name, confidence)
    trace_rule (number)
    trace_conclusion (name, name, name, confidence)
    trace_internal (name, name, name)
    wait
    stop_frage

/*  WISSENS-AKQUISITION */

    string_namelist (name,namelist)
    namelist_string (namelist, name)
    parse_rule (namelist, konsequenz, bedingungen)
    rule_string (konsequenz, bedingungen, name)
    aux_rule_string (bedingungen, name)
    edit_rule (number)
    edit_fact (number)
    update_number (number)
    update_facts (name, name, name, confidence)
    update_num_of_facts (number)
    edit_rules
    edit_facts
    edit_questions
    edit_question
    edit_conclusions
    edit_conclusion
    get_old_question (name, name, name, string)
    get_old_conclusion (name, name, name, confidence, confidence, string)
    list_quest (string)
    list_conc (string)
    get_command (char, symbol)
    do_edit_rules (symbol)
    do_edit_facts (symbol)
    get_cursor_command (char, symbol)
    display_rule (number)
    display_fact (number)
    help_edit

/*  DIALOG-KOMPONENTE */

    dialog
    execute (name)
    frage (name)
    use (konsequenz)
    not_used (konsequenz)
    clear_used
    unstop
    parse_question (name, namelist)
    get_answer (name, confidence)
    wort_con (name, confidence)
    con_wort (confidence, name)
    is_told (name, name, name, confidence)
    member (name, namelist)
    k_member (konsequenz, kons)
    conclude (name, name, name, confidence)

/*  ABLAUFSTEUERUNG */

    repeat
    init
```

```
  center (integer, String)
  load_dba (name)
  show_header
  clearmem
  clearfacts

goal
  init, dialog.

clauses

 /*****************************************************************/
 /*                      Inferenz-Maschine                        */
 /*****************************************************************/

  aux_wahr (_, _, _, _) if stop, !.
  aux_wahr (X, Y, Z, Con) if internal (Y),
    trace_internal (X, Y, Z),
    set_found,
    ist_wahr (X, Y, Z, Con),
    trace_conclusion (X, Y, Z, Con), !.
  aux_wahr (X, Y, Z, Con) if fact (_, [X, Y, Z], Con),
    set_found,
    trace_fact ("Ich weiß:", X, Y, Z, Con).
  aux_wahr (X, Y, Z, Con) if rule (R, dann (X, Y, Z), Bed, Con1),
    update_track (R),
    trace_rule (R),
    erfuellt (Bed, Con2),
    set_found,
    cut_track,
    Con2 >= 0,
    Con = Con1 * Con2.
  aux_wahr (X, Y, Z, Con) if found (no),
    question (X, Y, Z, _),
    is_told (X, Y, Z, Con),
    trace_fact ("Ich merke mir:", X, Y, Z, Con).
  aux_wahr (X, Y, Z, 0) if found (no),
    not (question  (X, Y, Z, _)),
    trace_fact ("Ich unterstelle:", X, Y, Z, 0).
  aux_wahr (_, _, _, -1).

  wahr (_, _, _, _) if stop, !.
  wahr (X, Y, Z, Con) if reset_found,
    findall (C, aux_wahr (X, Y, Z, C), Cons),
    max (Cons, Con), trace_conclusion (X, Y, Z, Con),
    conclude (X, Y, Z, Con).

  reset_found if retract (found (_)),
    asserta (found (no)), !.
  reset_found if asserta (found (no)), !.

  set_found if retract (found (_)),
    asserta (found (yes)), !.

  falsch (X, Y, Z, Con) if wahr (X, Y, Z, Con1),
    Con = 1 - Con1, !.

  erfuellt (_, 0) if stop, !.
  erfuellt ([], 1) if !.
  erfuellt ([wenn (X, Y, Z) | Rest], Con) if wahr (X, Y, Z, Con1) and
    getmincon (Mincon),
    Con1 >= Mincon,
    Con1 > 0,
    erfuellt (Rest, Con2) and
    min ([Con1, Con2], Con), !.
  erfuellt ([wenn_nicht (X, Y, Z) | Rest], Con) if falsch (X, Y, Z, Con1) and
    getmincon (Mincon),
    Con1 >= Mincon,
    Con1 > 0,
    erfuellt (Rest, Con2) and
    min ([Con1, Con2], Con), !.
  erfuellt (_, 0) if !.
```

```
 max ([X], X) if !.
 max ([X | Rest], X) if max (Rest, Y) and X > Y, !.
 max ([X | Rest], Y) if max (Rest, Y) and Y >= X, !.

 min ([X], X) if !.
 min ([X | Rest], X) if min (Rest, Y) and X < Y, !.
 min ([X | Rest], Y) if min (Rest, Y) and Y <= X, !.

getmincon (X) if mincon (X), !.
getmincon (0).

setmincon (X) if retract (mincon (_)), asserta (mincon (X)), !.
setmincon (X) if asserta (mincon (X)).

/*****************************************************************/
/*                    Erklärungs-Komponente                      */
/*****************************************************************/

 trace_on if retract (how_trace (_)),
   asserta (how_trace (on)), !.
 trace_on if asserta (how_trace (on)).

 trace_off if retract (how_trace (_)),
   asserta (how_trace (off)), !.
 trace_off if asserta (how_trace (off)).

 update_track (R) if
   retract (track (Spur)),
   asserta (track ([R|Spur])), !.
 update_track (R) if
   asserta (track ([R])), !.

 explain_why ([]) if shiftwindow (3), nl,
   write ("Ende des Erklärungs_Pfades."), nl, !.
 explain_why ([R|Rest]) if
   shiftwindow (3),
   clearwindow,
   nl, write ("Ich versuche Regel ", R, " anzuwenden:"), nl,
   rule (R, Kons, Bed, Con),
   rule_string (Kons, Bed, Regel),
   write (Regel), nl,
   write ("(W: ", Con, ")"), nl,
   write ("Weiter verfolgen (j/n)? "),
   readchar (C),
   C = 'j',
   nl,
   explain_why (Rest), !.
 explain_why (_).

 trace_fact (_, _, _, _, _) if how_trace (off), !.
 trace_fact (_, _, _, _, _) if stop, !.
 trace_fact (Message, X, Y, Z, Con) if shiftwindow (4),
   nl, write (Message), nl,
   write ("  ", X, " ", Y, " ", Z, " (", Con, ")"), nl,
   wait.

 trace_rule (_) if how_trace (off), !.
 trace_rule (_) if stop, !.
 trace_rule (R) if rule (R, Kons, Bed, Con),
   shiftwindow (4),
   nl, write ("Ich wende Regel Nr.", R, " an:"), nl,
   rule_string (Kons, Bed, Regel),
   write (Regel, " (", Con, ")"), nl,
   wait.

 trace_conclusion (_, _, _, _) if how_trace (off), !.
 trace_conclusion (_, _, _, _) if stop, !.
 trace_conclusion (X, Y, Z, Con) if shiftwindow (4),
   nl, write ("Ich folgere:"), nl,
   write ("  ", X, " ", Y, " ", Z, " (", Con, ")"), nl,
   wait.
```

```
trace_internal (_, _, _) if how_trace (off), !.
trace_internal (_, _, _) if stop, !.
trace_internal (X, Y, Z) if shiftwindow (4),
  nl, write ("Ich bewerte die Aussage"),
  nl, write ("     ", X, " ", Y, " ", Z, " intern."), nl,
  wait.

cut_track if retract (track ([_|Rest])),
  asserta (track (Rest)), !.
cut_track.

wait if readchar (C),
  char_int (C, 27),
  stop_frage, !.
wait.

stop_frage if asserta (stop).

/*****************************************************************/
/*                      Wissens-Akquisition                      */
/*****************************************************************/

string_namelist(S,[H|T]):-
    fronttoken(S,H,S1),!,string_namelist(S1,T).
string_namelist(_,[]).

namelist_string ([], "") if !.
namelist_string ([Y | Rest], X) if namelist_string (Rest, X1),
  concat (Y, " ", Y1),
  concat (Y1, X1, X), !.

parse_rule ([], _, []) if !.
parse_rule ([und, wenn, nicht, X, Y, Z | Rest], Kons,
            [wenn_nicht (X, Y, Z) | Bed]) if
  parse_rule (Rest, Kons, Bed), !.
parse_rule ([und, nicht, X, Y, Z | Rest], Kons,
            [wenn_nicht (X, Y, Z) | Bed]) if
  parse_rule (Rest, Kons, Bed), !.
parse_rule ([und, wenn, X, Y, Z | Rest], Kons,
            [wenn (X, Y, Z) | Bed]) if
  parse_rule (Rest, Kons, Bed), !.
parse_rule ([und, X, Y, Z | Rest], Kons,
            [wenn (X, Y, Z) | Bed]) if
  parse_rule (Rest, Kons, Bed), !.
parse_rule ([wenn, nicht, X, Y, Z | Rest], Kons,
            [wenn_nicht (X, Y, Z) | Bed])
  if parse_rule (Rest, Kons, Bed), !.
parse_rule ([wenn, X, Y, Z | Rest], Kons, [wenn (X, Y, Z) | Bed])
  if parse_rule (Rest, Kons, Bed), !.
parse_rule ([X, Y, Z | Rest], dann (X, Y, Z), Bed) if
  parse_rule (Rest, dann (X, Y, Z), Bed), !.

rule_string (dann (X, Y, Z), [wenn (X1, Y1, Z1)|Rest], R) if
  namelist_string ([X, Y, Z], S1),
  concat (S1, "\n   wenn ", S2),
  namelist_string ([X1, Y1, Z1], S3),
  concat (S2, S3, S4),
  aux_rule_string (Rest, S5),
  concat (S4, S5, R), !.
rule_string (dann (X, Y, Z), [wenn_nicht (X1, Y1, Z1)|Rest], R) if
  namelist_string ([X, Y, Z], S1),
  concat (S1, "\n   wenn nicht ", S2),
  namelist_string ([X1, Y1, Z1], S3),
  concat (S2, S3, S4),
  aux_rule_string (Rest, S5),
  concat (S4, S5, R), !.
rule_string (dann (X, Y, Z), [], R) if
  namelist_string ([X, Y, Z], S1),
  concat (S1, ".", R), !.

aux_rule_string ([], "") if !.
```

```
aux_rule_string ([wenn (X, Y, Z)|Rest], S) if
  namelist_string ([X, Y, Z], R),
  concat ("\n   und ", R, R1),
  aux_rule_string (Rest, S1),
  concat (R1, S1, S), !.
aux_rule_string ([wenn_nicht (X, Y, Z)|Rest], S) if
  namelist_string ([X, Y, Z], R),

  concat ("\n   und nicht ", R, R1),
  aux_rule_string (Rest, S1),
  concat (R1, S1, S), !.

edit_rule (R) if retract (rule (R, Kons, Bed, Con)),
  rule_string (Kons, Bed, Regel),
  str_int (R1, R),
  concat (" Regel ", R1, Title1),
  str_real (Con1, Con),
  concat (Title1, " (", Title2),
  concat (Title2, Con1, Title3),
  concat (Title3, ") ", Title),
  shiftwindow (8),
  removewindow,
  makewindow (8, 31, 7, Title, 12, 1, 11, 78),
  edit (Regel, Neuregel),
  clearwindow,
  write (Neuregel),
  nl, nl, write  ("Wahrheitswert für diese Regel: "),
  readreal (Newcon),
  shiftwindow (1),
  string_namelist (Neuregel, Regel2),
  parse_rule (Regel2, Kons1, Bed1),
  asserta (rule (R, Kons1, Bed1, Newcon)), !.

edit_rule (R) if
  not (rule (R, _, _, _)),
  str_int (R1, R),
  concat (" Regel ", R1, Title1),
  concat (Title1, " ", Title),
  shiftwindow (8),
  removewindow,
  makewindow (8, 31, 7, Title, 12, 1, 11, 78),
  edit ("", Regel),
  clearwindow,
  write (Regel),
  nl, nl, write ("Wahrheitswert für diese Regel: "),
  readreal (Con),
  clearwindow,
  update_number (R),
  string_namelist (Regel, Neuregel),
  parse_rule (Neuregel, Kons, Bed),
  asserta (rule (R, Kons, Bed, Con)), !.

edit_questions if repeat,
  edit_question.

edit_question if shiftwindow (7), clearwindow,
  write ("Fragetext für die Aussage ([Return] für Ende): "), nl,
  readln (Aussage),
  Aussage <> "",
  list_quest (Aussage),
  string_namelist (Aussage, [X, Y, Z]),
  get_old_question (X, Y, Z, Old),
  makewindow (11, 31, 7, Aussage, 12, 1, 11, 78),
  edit (Old, New),
  New <> "",
  asserta (question (X, Y, Z, New)),
  removewindow,
  !, fail.
edit_question.

list_quest ("list") if question (X, Y, Z, Text),
   concat (" ", X, T0),
```

```
      concat (T0, "  , T1),
      concat (T1, Y, T2),
      concat (T2, " ", T3),
      concat (T3, Z, T4),
      concat (T4, " ", Title),
      makewindow (13, 31, 7, Title, 12, 1, 11, 78),
      write (Text), readln (_),
      removewindow,
      fail.
    list_quest (_).

   edit_conclusions if repeat,
     edit_conclusion.

   edit_conclusion if shiftwindow (7), clearwindow,
     write ("Hinweis für Aussage ([Return] für Ende): "), nl,
     readln (Aussage),
     Aussage <> "",
     list_conc (Aussage),
     string_namelist (Aussage, [X, Y, Z]),
     write ("Hinweis ab Wahrheitswert: "), readreal (Min),
     write ("Hinweis bis Wahrheitswert: "), readreal (Max),
     get_old_conclusion (X, Y, Z, Min, Max, Old),
     makewindow (12, 31, 7, Aussage, 12, 1, 11, 78),
     edit (Old, New),
     removewindow,
     New <> "",
     asserta (conclusion (X, Y, Z, Min, Max, New)),
     !, fail.
   edit_conclusion.

   list_conc ("list") if conclusion (X, Y, Z, Min, Max, Text),
      concat (" ", X, T0),
      concat (T0, " ", T1),
      concat (T1, Y, T2),
      concat (T2, " ", T3),
      concat (T3, Z, T4),
      concat (T4, " (", T5),
      str_real (Min1, Min),
      concat (T5, Min1, T6),
      concat (T6, " - ", T7),
      str_real (Max1, Max),
      concat (T7, Max1, T8),
      concat (T8, ") ", Title),
      makewindow (13, 31, 7, Title, 12, 1, 11, 78),
      write (Text), readln (_),
      removewindow,
      fail.
    list_conc (_).

   get_old_question (X, Y, Z, Old) if retract (question (X, Y, Z, Old)), !.
   get_old_question (_, _, _, "").

   get_old_conclusion (X, Y, Z, Min, Max, Old) if retract (conclusion (X, Y,
Min, Max, Old)), !.
   get_old_conclusion (_, _, _, _, _, "").

   update_number (R) if rules (R1), R1 >= R, !.
   update_number (R) if retract (rules (_)), asserta (rules (R)), !.
   update_number (R) if asserta (rules (R)).

   update_facts (X, Y, Z, Con) if retract (facts (F)),
     F1 = F + 1,
     asserta (facts (F1)),
     asserta (fact (F1, [X, Y, Z], Con)), !.
   update_facts (X, Y, Z, Con) if asserta (facts (1)),
     asserta (fact (1, [X, Y, Z], Con)), !.

   update_num_of_facts (F) if retract (facts (_)), asserta (facts (F)), !.
   update_num_of_facts (F) if asserta (facts (F)).

   edit_rules if repeat,
```

```
  work_rule (R),
  display_rule (R),
  readchar (C),
  get_command (C, Com),
  do_edit_rules (Com),
  Com = quit.

display_rule (R) if rule (R, Kons, Bed, Con),
  rule_string (Kons, Bed, Altregel),
  shiftwindow (2),
  clearwindow,
  nl, write ("Regel Nr.", R, ":"), nl, nl,
  write (Altregel, "  (", Con, ")"), nl, !.
display_rule (R) if shiftwindow (2),
  clearwindow,
  nl, write ("Regel Nr.", R, ":"), nl.

do_edit_rules (last) if work_rule (1), beep, !.
do_edit_rules (last) if retract (work_rule (R)),
  R1 = R - 1,
  asserta (work_rule (R1)), !.
do_edit_rules (next) if rules (R),
  work_rule (W), W > R, beep, !.
do_edit_rules (next) if retract (work_rule (R)),
  R1 = R + 1,
  asserta (work_rule (R1)), !.
do_edit_rules (ed) if work_rule (R),
  edit_rule (R), !.
do_edit_rules (quit) if !.
do_edit_rules (help) if help_edit, !.
do_edit_rules (jump) if nl,
  write (" * Springen nach Regel Nr.: "),
  readint (R),
  rules (R1),
  R > 0,
  R < R1 + 2,
  retract (work_rule (_)),
  asserta (work_rule (R)), !.
do_edit_rules (explain) if execute ("e"), !.
do_edit_rules (_) if !.
```

```
edit_fact (F) if retract (fact (F, Faktum, Con)),
  namelist_string (Faktum, Altfakt),
  str_int (F1, F),
  concat (" Faktum ", F1, Title1),
  str_real (Con1, Con),
  concat (Title1, " (", Title2),
  concat (Title2, Con1, Title3),
  concat (Title3, ") ", Title),
  shiftwindow (8),
  removewindow,
  makewindow (8, 31, 7, Title, 12, 1, 11, 78),
  edit (Altfakt, Neufakt),
  clearwindow,
  write (Neufakt),
  nl, nl, write  ("Wahrheitswert fUr dieses Faktum: "),
  readreal (Newcon),
  shiftwindow (1),
  string_namelist (Neufakt, Neufakt1),
  assertz (fact (F, Neufakt1, Newcon)), !.

edit_fact (F) if
  str_int (F1, F),
  concat (" Faktum ", F1, Title1),
  concat (Title1, " ", Title),
  shiftwindow (8),
  removewindow,
  makewindow (8, 31, 7, Title, 12, 1, 11, 78),
  edit ("", Faktum),
  clearwindow,
  write (Faktum),
  nl, nl, write ("Wahrheitswert fUr dieses Faktum: "),
  readreal (Con),
  clearwindow,
  update_num_of_facts (F),
  string_namelist (Faktum, Neufakt),
  assertz (fact (F, Neufakt, Con)).

edit_facts if repeat,
  work_fact (F),
  display_fact (F),
  readchar (C),
  get_command (C, Com),
  do_edit_facts (Com),
  Com = quit.

do_edit_facts (last) if work_fact (1), beep, !.
do_edit_facts (last) if retract (work_fact (F)),
  F1 = F - 1,
  asserta (work_fact (F1)), !.
do_edit_facts (next) if facts (F),
  work_fact (W), W > F, beep, !.
do_edit_facts (next) if retract (work_fact (F)),
  F1 = F + 1,
  asserta (work_fact (F1)), !.
do_edit_facts (ed) if work_fact (F),
  edit_fact (F), !.
do_edit_facts (quit) if !.
do_edit_facts (help) if help_edit, !.
do_edit_facts (jump) if nl,
  write (" * Springen nach Faktum Nr.: "),
  readint (F),
  facts (F1),
  F > 0,
  F < F1 + 2,
  retract (work_fact (_)),
  asserta (work_fact (F)), !.
do_edit_facts (explain) if execute ("e"), !.
do_edit_facts (_) if !.
```

```
    display_fact (F) if fact (F, Faktum, Con),
      namelist_string (Faktum, Altfakt),
      shiftwindow (2),
      clearwindow,
      nl, write ("Faktum Nr.", F, ":"), nl, nl,
      write (Altfakt, "  (", Con, ")"), nl, !.
    display_fact (F) if shiftwindow (2),
      clearwindow,
      nl, write ("Faktum Nr.", F, ":"), nl.

    get_command (C, Com) if char_int (C, 0),
      readchar (C1),
      get_cursor_command (C1, Com), !.
    get_command ('?', help) if !.
    get_command ('#', jump) if !.
    get_command (C, ed) if char_int (C, 13), !.
    get_command (C, quit) if char_int (C, 27), !.
    get_command ('e', explain) if !.

    get_cursor_command ('H', last) if !.
    get_cursor_command ('P', next) if !.
    get_cursor_command ('M', jump) if !.
    get_cursor_command ('K', ed) if !.
    get_cursor_command ('O', quit) if !.

    help_edit if shiftwindow (9),
      clearwindow,
      nl,
      char_int (U, 24),
      char_int (D, 25),
      char_int (L, 26),
      char_int (R, 27),
      write ("  Taste       Funktion"), nl,
      write ("-------------------------------------------------------------------
----------"), nl,
      write ("    ", U, "          Zurück                    End, Escape   Funktion
 beenden"), nl,
      write ("    ", L, ", #       Springe nach Nr.          Return, ", R, "     Ei
ntrag Ändern"), nl,
      write ("    ", D, "          Weiter").

   /*****************************************************************/
   /*                        Dialog-Komponente                      */
   /*****************************************************************/

    dialog if repeat,
      shiftwindow (1),
      write (")"),
      clear_used,
      unstop,
      readln (Command),
      execute (Command),
      fail.

    execute ("r") if edit_rules, !.
    execute ("w") if retract (how_trace (on)),
      asserta (how_trace (off)),
      write (" * Wie-Trace ausgeschaltet."),
      nl, !.
    execute ("w") if retract (how_trace (off)),
      asserta (how_trace (on)),
      write (" * Wie-Trace eingeschaltet."),
      nl, !.
    execute ("?") if shiftwindow (6), clearwindow,
      write ("  * - Befragung beginnen"), nl,nl,
      write ("  ? - Eingabemöglichkeiten                w - Ein/Ausschalten des Wie
-Trace"), nl,
      write ("  r - Ändern von Regeln                   s - Speichern der Wissensba
sis"), nl,
      write ("  f - Ändern von Fakten                   b - Beenden der Konsultatio
n ), nl,
      write ("  l - Löschen aller Fakten                m - freier Speicherplatz"),
```

```
nl,
      write ("  i - Interne Relationen sehen          e - Erläuterungen sehen/änd
ern"), nl,
      write ("  t - Fragetexte erstellen              h - Hinweise erstellen"), n
l,
      write ("  g - Grenz-Wahrheitswert festlegen     z - Ziel definieren"), !.
    execute ("z") if ziel (X),
      write ("  Altes Ziel: ", X), nl,
      write ("  Neues Ziel ([Return] = keine Änderung): "), readln (Z),
      Z <> "",
      retract (ziel (_)), asserta (ziel (Z)), !.
    execute ("z") if not (ziel (_)),
      write ("  Ziel: "), readln (Z),
      asserta (ziel (Z)), !.
    execute ("*") if retract (was_answer), fail.
    execute ("*") if ziel (Z), not (frage (Z)), !.
    execute ("g") if getmincon (Mincon),
      write (" Der alte Grenz-Wahrheitswert ist ", Mincon), nl,
      write (" Neuer Grenzwert ([Return] = keine Änderung): "), readln (Newmin),
      Newmin <> "",
      str_real (Newmin, Newmincon),
      setmincon (Newmincon), !.
    execute ("t") if edit_questions, !.
    execute ("h") if edit_conclusions, !.
    execute ("m") if nl,
      storage (_, H, _),
      write (" * Freier Speicherplatz: ", H, " Bytes."), nl, nl, !.
    execute ("l") if nl,
      write (" * Fakten wirklich löschen (j) ? "),
      readchar (C),
      C = 'j',
      nl,
      clearfacts, !.
    execute ("f") if edit_facts, !.
    execute ("s") if write (" * Abspeichern der Wissensbasis unter: "),
      readln (Name),
      concat (Name, ".dba", Filename),
      retract (work_rule (R)),
      retract (work_fact (F)),
      retract (how_trace (T)),
      save (Filename),
      asserta (work_rule (R)),
      asserta (work_fact (F)),
      asserta (how_trace (T)),
      write (" ...gespeichert."), nl, !.
    execute ("b") if write (" * Konsultation wirklich beenden (j) ? "),
      readchar (Input), nl,
      Input = 'j',
      clearmem,
      exit.
    execute ("i") if shiftwindow (10),
      is_internal (X), write (X, "    "), fail.
    execute ("e") if shiftwindow (5),
      show_header,
      execute ("?"), !.
    execute (_) if retract (was_answer), fail.
    execute (I) if str_len (I, L), L > 1, not (frage (I)), !.
    execute (_).

    frage (_) if stop, !.
    frage (Frage) if parse_question (Frage, [X, Y, Z]),
      internal (Y),
      trace_internal (X, Y, Z),
      ist_wahr (X, Y, Z, Con),
      trace_conclusion (X, Y, Z, Con),
      shiftwindow (1),
      getmincon (Mincon),
      Con >= Mincon,
      con_wort (Con, Wort),
      write ("--> ", X, " ", Y, " ", Wort, " ", Z, " (", Con, ")"), nl, !.
    frage (Frage) if
      parse_question (Frage, [X, Y, Z]),
```

```
      rule (_, dann (X, Y, Z), _, _),
      not_used (dann (X, Y, Z)),
      use (dann (X, Y, Z)),
      not (stop),
      wahr (X, Y, Z, Con),
      shiftwindow (1),
      not (stop),
      getmincon (Mincon),
      Con >= Mincon,
      con_wort (Con, Wort),
      write ("--> ", X, " ", Y, " ", Wort, Z, " (", Con, ")"), nl,
      asserta (was_answer),
      fail.
  frage (_) if stop, !.
  frage (_) if not (was_answer), getmincon (Mincon),
    write ("Ich finde keine Lösung mit Wahrheitswert größer/gleich ", Mincon, ".
"), nl.

    use (K) if retract (used (Kons)),
      asserta (used ([K|Kons])), !.
    use (K) if asserta (used ([K])), !.

    not_used (K) if used (Kons),
      k_member (K, Kons), !, fail.
    not_used (_) if !.

    clear_used if retract (used (_)), !.
    clear_used.

    unstop if retract (stop).
    unstop.

    parse_question (Frage, [_, Y, _]) if
      string_namelist (Frage, ["X", Y, "Y"]), !.
    parse_question (Frage, [_, Y, Z]) if
      string_namelist (Frage, ["X", Y, Z]), !.
    parse_question (Frage, [X, Y, _]) if
      string_namelist (Frage, [X, Y, "X"]), !.
    parse_question (Frage, [X, Y, Z]) if
      string_namelist (Frage, [X, Y, Z]), !.

    get_answer ("?", Con) if shiftwindow (6),
      clearwindow,
      write ("  warum : Erklärung der Frage"), nl, nl,
      write ("  ja (1.0), sicher (1.0), höchst wahrscheinlich (0.95), "), nl,
      write ("  sehr wahrscheinlich (0.9), wahrscheinlich (0.8), vermutlich (0.7
)"), nl,
      write ("  möglicherweise (0.6), vielleicht (0.5), weiß nicht (0.5),"), nl,
      write ("  möglicherweise nicht (0.4), vermutlich nicht (0.3), wahrscheinli
ch"), nl,
      write ("  nicht (0.2), sehr wahrscheinlich nicht (0.1), nein (0), sicher n
icht (0)"), nl, nl,
      write ("  ... oder eine Dezimalzahl zwischen 0 und 1."),
      shiftwindow (1), nl,
      write ("  Ihre Eingabe: "), readln (Newanswer),
      get_answer (Newanswer, Con), !.
    get_answer ("warum", Con) if track (Spur),
      explain_why (Spur),
      shiftwindow (1),
      write ("  Ihre Eingabe: "),
      readln (Answer), nl,
      wort_con (Answer, Con), !.
    get_answer (Wort, Con) if wort_con (Wort, Con), nl, !.

    wort_con ("ja", 1) if !.
    wort_con ("sicher", 1) if !.
    wort_con ("höchst wahrscheinlich", 0.95) if !.
    wort_con ("sehr wahrscheinlich", 0.9) if !.
    wort_con ("wahrscheinlich", 0.8) if !.
    wort_con ("vermutlich", 0.7) if !.
    wort_con ("möglicherweise", 0.6) if !.
    wort_con ("vielleicht", 0.5) if !.
```

```
wort_con ("weiß nicht", 0.5) if !.
wort_con ("möglicherweise nicht", 0.4) if !.
wort_con ("vermutlich nicht", 0.3) if !.
wort_con ("wahrscheinlich nicht", 0.2) if !.
wort_con ("sehr wahrscheinlich nicht", 0.1) if !.
wort_con ("nein", 0) if !.
wort_con ("sicher nicht", 0) if !.
wort_con (X, Con) if str_real (X, Con), !.
wort_con (_, Con) if
  write ("  Das habe ich nicht verstanden. Neue Eingabe: "),
  readln (Wort),
  get_answer (Wort, Con).

con_wort (1, "") if !.
con_wort (C, "höchst wahrscheinlich ") if C >= 0.95, !.
con_wort (C, "sehr wahrscheinlich ") if C >= 0.9, !.
con_wort (C, "wahrscheinlich ") if C >= 0.8, !.
con_wort (C, "vermutlich ") if C >= 0.7, !.
con_wort (C, "möglicherweise ") if C >= 0.6, !.
con_wort (C, "vielleicht ") if C >= 0.5, !.
con_wort (C, "möglicherweise nicht ") if C >= 0.4, !.
con_wort (C, "vermutlich nicht ") if C >= 0.3, !.
con_wort (C, "wahrscheinlich nicht ") if C >= 0.2, !.
con_wort (C, "sehr wahrscheinlich nicht ") if C >= 0.1, !.
con_wort (C, "höchst wahrscheinlich nicht ") if C > 0, !.
con_wort (0, "nicht ").

is_told (X, Y, Z, Con) if shiftwindow (1),
  question (X, Y, Z, Question),
  nl, write (Question, " "),
  readln (Answer),
  get_answer (Answer, Con),
  update_facts (X, Y, Z, Con), !.

member (_, []) if !, fail.
member (X, [X | _]).
member (X, [_ | Rest]) if member (X, Rest).

k_member (_, []) if !, fail.
k_member (X, [X | _]).
k_member (X, [_ | Rest]) if k_member (X, Rest).

conclude (X, Y, Z, Con) if was_con (X, Y, Z, Con), !.
conclude (X, Y, Z, Con) if conclusion (X, Y, Z, Min, Max, String),
  Con >= Min,
  Con <= Max,
  shiftwindow (15),
  display (String),
  asserta (was_con (X, Y, Z, Con)), !.
conclude (_, _, _, _).


/*****************************************************************/
/*                        Ablaufsteuerung                        */
/*****************************************************************/

init if trace_on,
  makewindow (5, 7, 7, " * MOSES V 1.0 * ", 0, 0, 24, 80),
  shiftwindow (5),
  center (3, "Modulare Shell für Experten-Systeme"),
  center (4, "Version 1.0 vom 12.5.1987"),
  center (8, "(c) 1987"),
  center (9, "by Karl-Ludwig von Wendt"),
  cursor (16, 26),
  write ("Name der Wissensbasis: "),
  readln (Name),
  concat (Name, ".dba", Filename),
  load_dba (Filename),
  show_header,
  concat (" ", Name, Title1),
  concat (Title1, " ", Title),
  makewindow (15, 79, 7, " Hinweis ", 12, 1, 11, 78),
```

```
  makewindow (9, 31, 7, " Hilfe ", 1, 1, 11, 78),
  makewindow (1, 112, 7, Title, 1, 1, 11, 78),
  makewindow (8, 112, 7, "", 12, 1, 11, 78),
  makewindow (6, 31, 7, " Eingabemöglichkeiten ", 12, 1, 11, 78),
  makewindow (7, 112, 7, " Dialog-Text ", 12, 1, 11, 78),
  makewindow (2, 112, 7, " Wissensbasis ", 12, 1, 11, 78),
  makewindow (4, 31, 7, " Wie ", 12, 1, 11, 78),
  makewindow (3, 31, 7, " Warum ", 12, 1, 11, 78),
  makewindow (10, 112, 7, " Interne Relationen ", 12, 1, 11, 78),
  execute ("?"),
  shiftwindow (1), nl,
  execute ("w"),
  execute ("m"),
  asserta (work_rule (1)),
  asserta (work_fact (1)).

load_dba (Filename) if consult (Filename), !.
load_dba (_) if asserta (rules (0)), asserta (facts (0)).

show_header if retract (header (Text)),
  edit (Text, Neutext),
  asserta (header (Neutext)), !.
show_header if edit (" *** Erläuterungen ***", Text),
  asserta (header (Text)), !.

center (Y, String) if str_len (String, L),
  X = (78 - L)/2,
  cursor (Y, X),
  write (String).

repeat.
repeat if repeat.

clearmem if retract (_), fail.
clearmem if not (retract (_)), !.

clearfacts if retract (was_con (_, _, _, _)), fail.
clearfacts if retract (fact (_, _, _)), fail.
clearfacts if not (fact (_, _, _)),
  retract (facts (_)),
  asserta (facts (0)), !.

mincon (0).
```

Anhang C

Prolog-Listing:

Simulationsmodell zur Liquiditätsprognose

```
/*************************************************/
/* LIQUID - MOSES-Modul zur LiquiditÄtsprognose */
/* auf der Basis einer Monte-Carlo-Simulation   */
/*************************************************/
project "moses"
global domains

    namelist = reference name*
    name  = reference string
    konsequenz = reference dann (name, name, name)
    kons = reference konsequenz*
    bedingung  = reference  wenn (name, name, name);
                            wenn_nicht (name, name, name)
    bedingungen = reference bedingung*
    confidence = reference real
    cons = reference confidence*
    number = reference integer
    numbers = reference number*


  database
    context (name, namelist)
    rule (number, konsequenz, bedingungen, confidence)
    fact (number, namelist, confidence)
    rules (number)
    facts (number)
    question (name, name, name, string)
    conclusion (name, name, name, confidence, confidence, string)
    was_con (name, name, name, confidence)
    track (numbers)
    how_trace (symbol)
    found (symbol)
    work_rule (number)
    work_fact (number)
    header (string)
    used (kons)
    mincon (confidence)
    stop
    was_answer
    ziel (name)
    data (integer, name, real)

  global predicates
    is_internal (name) - (o)
    internal (name) - (i)
    ist_wahr (name, name, name, confidence) - (i, i, i, o)

predicates
  og (integer, real)
  ug (integer, real)
  ew (integer, real)
  wert (integer, name, real)
  wert1 (integer, name, real)
  get_value (integer, real, real)
  liquid (integer, real)
  versuch (integer, integer)
  update_facts (number)

 clauses
  internal ("LiquiditÄt_ist").
  is_internal ("LiquiditÄt_ist").
```

```
  ist_wahr (_, "Liquidität_ist", _, Con) if
    V = 100,
    makewindow (20, 31, 7, " Hilfe ", 12, 1, 11, 73),
    write ("Ich versuche nun, zu prüfen, ob ihre Finanzplanung ausreicht, um die
 Liqui-"), nl,
    write ("dität Ihres Unternehmens in den ersten fünf Jahren zu gewährleisten.
 Dazu"), nl,
    write ("werde ich mehrere Male mögliche Zukunftsentwicklungen durchspielen,
indem"), nl,
    write ("ich die zukünftigen Umsätze innerhalb der von Ihnen angegebenen Gren
zen"), nl,
    write ("zufällig auswähle. Bitte haben Sie etwas Geduld, die Berechnungen da
uern"), nl,
    write ("einige Zeit. Versuchen Sie bitte, die Werte, die ich erfrage, möglic
hst"), nl,
    write ("genau und realistisch zu schätzen. Wenn die Prognose abläuft, sehen
Sie"), nl,
    write ("bei jedem Versuch ein P wenn Ihr Unternehmen in Schwierigkeiten geko
mmen"), nl,
    write ("ist, andernfalls einen Punkt."),
    makewindow (21, 112, 7, " Liquiditätsprognose ", 1, 1, 11, 78),
    versuch (V, Pleiten),
    nl, write ("Von ", V, " Versuchen sind Sie ", Pleiten, " mal in Schwierigkei
ten gekommen."), nl,
    write ("Bitte drücken Sie jetzt RETURN."), readln (_),
    removewindow,
    shiftwindow (20),
    removewindow,
    Pleiten < (V / 2),
    Con = ((V / 2) - Pleiten) * 2 / V, !.


  get_value (Jahr, Input, Output) if
    og (Jahr, OG), ug (Jahr, UG), ew (Jahr, EW),
    ZC = (EW - UG) /(OG - UG),
    Input > ZC,
    Wurzel = ((OG - EW) * (OG - EW)) - ((OG - EW) * (OG - UG) * (Input - ZC)),
    Output = OG - sqrt (Wurzel), !.
  get_value (Jahr, Input, Output) if
    og (Jahr, OG), ug (Jahr, UG), ew (Jahr, EW),
    Wurzel = (EW - UG) * (OG - UG) * Input,
    Output = UG + sqrt (Wurzel), !.

  liquid (0, AB) if
    fact (_, ["Anfangs_Liquidität", "hat_wert", Ls], _),
    str_real (Ls, AB), !.
  liquid (0, AB) if
    nl, write (" Wie hoch ist Ihre Liquiditätsreserve zu Beginn? "),
    readreal (AB),
    str_real (Ls, AB),
    update_facts (F1),
    asserta (fact (F1, ["Anfangs_Liquidität", "hat_wert", Ls], 1)), !.
  liquid (Jahr, Ende) if
    Jahr1 = Jahr - 1,
    liquid (Jahr1, Anfang),
    random (Z),
    get_value (Jahr, Z, Umsatz),
    wert (Jahr, "Variable_Kosten", KV), wert (Jahr, "Zinsen", Zi),
    wert (Jahr, "Tilgungen", Til),
    wert (Jahr, "Betriebskosten", KF), wert (Jahr, "Privatentnahmen", E),
    random (Z2),
    Ende = Anfang + (1 - Kv) * Umsatz - ((5 + Z2) / 5) * KF - Zi - Til - E,
    Ende > 0, !.
```

```
og (Jahr, OG) if str_int (S1, Jahr),
  concat ("Jahr_", S1, S2),
  fact (_, [S2, "Umsatz_ist_höchstens", OGS], _),
  str_real (OGS, OG), !.
og (Jahr, OG) if nl,
  write (" Wie könnte der Umsatz in Jahr ", Jahr,
         " im günstigsten Fall sein? "),
  readreal (OG),
  str_int (JS, Jahr),
  concat ("Jahr_", Js, S1),
  str_real (S2, OG),
  update_facts (F1),
  asserta (fact (F1, [S1, "Umsatz_ist_höchstens", S2], 1)), !.

ug (Jahr, UG) if str_int (S1, Jahr),
  concat ("Jahr_", S1, S2),
  fact (_, [S2, "Umsatz_ist_mindestens", UGS], _),
  str_real (UGS, UG), !.
ug (Jahr, UG) if nl,
  write (" Wie hoch ist der Umsatz in Jahr ", Jahr,
         " im schlimmsten Fall mindestens? "),
  readreal (UG),
  str_int (JS, Jahr),
  concat ("Jahr_", Js, S1),
  str_real (S2, UG),
  update_facts (F1),
  asserta (fact (F1, [S1, "Umsatz_ist_mindestens", S2], 1)), !.

ew (Jahr, EW) if str_int (S1, Jahr),
  concat ("Jahr_", S1, S2),
  fact (_, [S2, "Umsatz_ist_wahrscheinlich", EWS], _),
  str_real (EWS, EW), !.
ew (Jahr, EW) if nl,
  write (" Welchen Umsatz halten Sie im Jahr ", Jahr,
         " für am wahrscheinlichsten? "),
  readreal (EW),
  str_int (JS, Jahr),
  concat ("Jahr_", Js, S1),
  str_real (S2, EW),
  update_facts (F1),
  asserta (fact (F1, [S1, "Umsatz_ist_wahrscheinlich", S2], 1)), !.

wert (Jahr, Name, Wert) if data (Jahr, Name, Wert), !.
wert (Jahr, Name, Wert) if
  wert1 (Jahr, Name, Wert),
  asserta (data (Jahr, Name, Wert)), !.
```

```
  wert1 (_, "Variable_Kosten", Wert) if
    fact (_, ["Variable_Kosten", "Anteil_am_Umsatz_ist", S1], _),
    str_real (S1, Wert), !.
  wert1 (_, "Variable_Kosten", Wert) if nl,
    write (" Wie hoch sind ihre variablen Kosten (Waren-, Materialeinsatz etc.)
), nl,
    write (" in Prozent vom Umsatz? "), readreal (KvPro),
    Wert = KvPro / 100,
    str_real (S1, Wert),
    update_facts (F1),
    asserta (fact (F1, ["Variable_Kosten", "Anteil_am_Umsatz_ist", S1], 1)), !.
  wert1 (Jahr, Symbol, Wert) if
    str_int (JS, Jahr),
    concat ("Jahr_", JS, S1),
    concat (Symbol, "_sind", S2),
    fact (_, [S1, S2, S3], _),
    str_real (S3, Wert), !.
  wert1 (Jahr, Symbol, Wert) if nl,
    write (" Wie hoch schätzen Sie die ", Symbol, " in Jahr ", Jahr, " ? "),
    readreal (Wert),
    str_int (JS, Jahr),
    concat ("Jahr_", JS, S1),
    concat (Symbol, "_sind", S2),
    str_real (S3, Wert),
    update_facts (F1),
    asserta (fact (F1, [S1, S2, S3], 1)), !.

  versuch (0, 0) if !.
  versuch (Nummer, Pleiten) if
    liquid (5, _),
    write (".",
    N1 = Nummer - 1,
    versuch (N1, Pleiten), !.
  versuch (Nummer, Pleiten1) if
    N1 = Nummer - 1,
    write ("P"),
    versuch (N1, Pleiten),
    Pleiten1 = Pleiten + 1, !.

  update_facts (X1) if retract (facts (X)), X1 = X + 1, asserta (facts (X1)), !.
  update_facts (1) if asserta (facts (1)), !.
```

Literaturverzeichnis

(1) Adam, D., Kurzlehrbuch Planung, Wiesbaden 1980.

(2) Appelrath, H. J., Die Erweiterung von DB- und IR Systemen zu Wissensbasierten Systemen, in: Nachrichten für Dokumentation, 36.Jg., 1985, Nr.1, S.13 ff.

(3) Bell, M.Z., Why Expert Systems Fail, in: Journal of the Operational Research Society, Vol.36, 1985, No.7, S.613 ff.

(4) Bellmann, R.E., L.A. Zadeh, Decision Making in a Fuzzy Environment, in: Management Science, Vol. 17-B, 1970, S.141 ff.

(5) Beutel, P., W. Schubö, SPSS 9, 4.Aufl., Stuttgart, New York 1983.

(6) Bleymüller, J., G. Gehlert, H. Gülicher, Statistik für Wirtschaftswissenschaftler, München 1979.

(7) Bobillier, P.A., B.C. Kahan, R.A. Probst, Simulation with GPSS and GPSS V, Englewood Cliffs 1976.

(8) Bouissone, P., Plausible Reasoning, in: S. Shapiro (Hrsg.), Encyclopedia of AI, New York u.a. 1987, S.854 ff.

(9) Buchanan, B.G., Expert systems: working systems and the research literature, in: Expert Systems, Vol.3, 1986, No.1, S.32 ff.

(10) Buchanan, B.G., E.A. Feigenbaum, Dendral and Meta-Dendral: the applications dimension, in: Artificial Intelligence, Vol. 11, 1978, S.5 ff.

(11) Buchanan, B.G. u.a., Constructing an Expert System, in: Hayes Roth, F., D.A. Waterman, D.B. Lenat (Hrsg.), Building Expert Systems, Reading (Mass.) u.a. 1983.

(12) Büttner, U. u.a., Stand des Expertensystems BILEX zur Bilanzanalyse, Arbeitspapiere der Informatik-Forschungsgruppe VIII der Friedrich-Alexander-Universität Erlangen Nürnberg, Hrsg. P. Mertens, Erlangen 1987.

(13) Campbell, A.N. u.a., Recognition of a hidden mineral deposit by an artificial intelligence program, in: Science, Vol. 217, 1982, No. 9, S.927 ff.

(14) Clocksin, W.F., C.S. Mellish, Programming in Prolog, Berlin, Heidelberg, New York 1981.

(15) Cooper, P., Expert Systems in Management Science, in: Future Generations Computer Systems, H.2, 1986, S.217 ff.

(16) Coombs, M.J., Expert Systems - Concepts and Examples, Manchester 1984.

(17) Dittrich, H., Wege und Tips zur Existenzgründung, München 1985.

(18) Doukidis, G.I., R.J.Paul, Research into Expert Systems to Aid Simulation Model Formulation, in: Journal of the Operational Research Society, Vol. 36, 1985, No. 4, S.319 ff.

(19) Dräger, U. u.a., Stand des Expertensystems zur GuV Analyse, Arbeitspapiere der Informatik-Forschungsgrup pe VIII der Friedrich-Alexander-Universität Erlangen Nürnberg, Hrsg. P. Mertens, Erlangen 1987.

(20) Ernst, M.L., H. Ojha, Business Applications of Artificial Intelligence Knowledge Based Expert Systems, in: Future Generation Computer Systems, H. 2, 1986, S.173 ff.

(21) Fohmann, L., Wissenserwerb und maschinelles Lernen, in: S. Savory (Hrsg.), Künstliche Intelligenz und Expertensysteme, 2.Aufl., München, Wien 1985.

(22) Fordyce, K., P. Norden, G. Sullivan, Artificial In telligence and the Management Science Practitioner: Links between Operations Research and Expert Systems, in: Interfaces, Vol. 17, H. 7/8, 1987, S.34 ff.

(23) Fox, M.S., S.F.Smith, ISIS - a knowledge-based sy stem for factory scheduling, in: Expert Systems, Vol.1, 1984, Nr.1, S.25 ff.

(24) Fuchi, K., Aiming for Knowledge Information Proces sing Systems, in: Moto-Oka, K. (Hrsg.), Fifth Gene ration Computer Systems, 3.Aufl., Amsterdam, New York, Oxford 1985, S.116 ff.

(25) Ganoe, F.J., Design and Evaluation of a Prototype Fuzzy-Logic-Based Knowledge System for Financial Ratio Analysis, Diss., Georgia State University 1986.

(26) Gaschnig, J., Prospector: An expert system for mi neral exploration, in: D. Michie (Hrsg.), Introduc tory Readings in Expert Systems, New York, London, Pa ris 1982, S.47 ff.

(27) Gaschnig, J. u.a., Evaluation of Expert Systems: Issues and Case Studies, in: Hayes Roth, F., D.A. Waterman, D.B. Lenat (Hrsg.), Building Expert Systems, Reading (Mass.) u.a. 1983, S.241 ff.

(28) Geller, J., Smalltalk, in: S. Shapiro (Hrsg.), Encyclopedia of AI, Vol.2, New York u.a. 1987, S.1049.

(29) Giannesini, F. u.a., Prolog, o.O. (Addison Wesley Deutschland GmbH) 1986.

(30) Guntzer, U. u.a., Filialanalyse bei der Hypo-Bank, in: Expertensysteme, State of the Art 1/86, Hrsg. P. Schnupp, München 1986, S.38 ff.

(31) Hahn, U., Expertensysteme als intelligente Informationssysteme - Konzepte für die funktionale Erweiterung des Information Retreival, in: Nachrichten für Dokumentation, 36. Jg., 1985, Nr.1, S.2 ff.

(32) Hammond, P., micro-Prolog for Expert Systems, in: Clark, C.L., F.G. McCabe, micro-Prolog: Programming in Logic, Englewood Cliffs u.a. 1984.

(33) Harmon, King, Expertensysteme in der Praxis, München, Wien 1986.

(34) Heimsoeth Software GmbH & Co. (Hrsg.), Turbo Prolog Handbuch, München 1986.

(35) Hewett, J., S. Timms, G. d'Annale, Commercial Expert Systems in Europe, Hrsg. Ovon Ltd., London 1986.

(36) Hofmann, M., Existenzgründung, 3.Aufl., Heidelberg 1986.

(37) Honig, G., So mache ich mich selbständig, 8.Aufl., München 1984.

(38) Iudica, N.R., GUMMEX - ein Expertensystem zur Generierung von Arbeitsplänen für die Fertigung, in: Nachrichten fur Dokumentation, 36. Jg., 1985, Nr. 1, S.22 ff.

(39) Jackson, P., Introduction to expert systems, Workingham (England) u.a. 1986.

(40) Kirschbaum, G., W. Naujoks, Erfolgreich in die berufliche Selbständigkeit, Freiburg i.Br. 1985.

(41) Koch, U., S. Heuer, Die Expertensystem-Shell DONALD, Schriftenreihe des Instituts für Angewandte Mathematik und Informatik der Universität Münster, H. 8/86 I, Münster 1986.

(42) Koreimann, D.S., Methoden und Organisation von Management-Informations-Systemen, Berlin, New York 1971.

(43) Lehman, E., Expertensysteme - überblick über den aktuellen Entwicklungsstand (1983), Hrsg. Siemens AG, München 1983.

(44) Liebermann, H., Object Oriented Languages, in: S. Shapiro (Hrsg.), Encyclopedia of AI, New York u.a. 1987, S. 452 ff.

(45) Luconi, F.L., T.W. Malone, M.S. Scott Morton, Expert Systems: The Next Challenge for Managers, in: Information Management, H. 3, 1986, S.6 ff.

(46) Maida, A.S., Frame Theory, in: S. Shapiro (Hrsg.), Encyclopedia of AI, New York u.a. 1987, S.302 ff.

(47) McDermott, J., R1: A Rule-Based Configurer of Computer Systems, in: Artificial Intelligence, Vol. 19, 1982, S.39 ff.

(48) McDermott, J., XSEL: a computer sales person´s assistant, in: Hayes, J.E., D. Michie, Y.-H. Pao, Machine Intelligence 10, Chichester 1982, S.325 ff.

(49) Meffert, H., Marketing, 5.Aufl., Wiesbaden 1980.

(50) Meili, R., Intelligenz, in: W. Arnold, H. J. Eysenck, R.Meili (Hrsg.), Lexikon der Psychologie, Bd. 2, Freiburg, Basel, Wien 1980, Sp.997 ff.

(51) Mertens, P., Einflüsse der EDV auf die Weiterentwicklung des betrieblichen Rechnungswesens, in: 4. Saarbrücker Arbeitstagung "Rechnungswesen und EDV", Hrsg. W. Kilger, A.W. Scheer, Würzburg, Wien 1983.

(52) Mertens, P., K. Allgeyer, Künstliche Intelligenz in der Betriebswirtschaft, in: Zeitschrift für Betriebswirtschaft, 53. Jg. 1983, H. 7, S.686 ff.

(53) Mertens, P., K. Allgeyer, H. Däs, Betriebliche Expertensysteme in deutschsprachigen Ländern, in: Zeitschrift für Betriebswirtschaft, 56.Jg., 1986, S.905 ff.

(54) Mertens, P., E. Plattfaut, Informationstechnik als strategische Waffe, in: Information Management, Nr. 2, 1986, S.6 ff.

(55) Mescheder, B., Prolog - Implementierungssprache der künstlichen Intelligenz, in: S. Savory (Hrsg.), Künstliche Intelligenz und Expertensysteme, München, Wien 1985, S.91 ff.

(56) Morgado, E., Meta-Knowledge, Meta-Rules and Meta-Reasoning, in: S.C. Shapiro (Hrsg.), Encyclopedia of AI, Vol.1, New York u.a. 1987, S.432 ff.

(57) Moto-Oka, T. (Hrsg.), Fifth Generation Computer Systems, 3. Aufl., Amsterdam, New York, Oxford 1985.

(58) Müller Merbach, H., Operations Research, 3. Aufl., München 1973.

(59) Negoita, C.V., Expert Systems and Fuzzy Systems, Menlo Park (Cal.) u.a. 1985.

(60) Newell, A., H.A. Simon, Heuristic Problem Solving: The next Advantage in Operations Research, in: Operations Research, Vol. 6, 1958, S.1 ff.

(61) Nieschlag, R., E. Dichtl, H. Hörschgen, Marketing, 14.Aufl., Berlin 1985.

(62) Noelke, U., Das Wesen des Knowledge Engineering, in: S. Savory (Hrsg.), Künstliche Intelligenz und Expertensysteme, München, Wien 1985, S.109 ff.

(63) o.V., Alternative mit Vorbehalt, in: Computer Persönlich, H. 14, 1987, S.36 ff.

(64) o.V., Experten über Expertensysteme: Wunsch und Wirklichkeit, in: Computer Magazin, H. 7/8, 1987, S. 12 ff.

(65) o.V., Speichertechnologie der Zukunft, in: Computer Persönlich, H. 14, 1987, S.38 ff.

(66) O'Keefe, R.M., Belton, V., Ball, T., Experiences with using Expert Systems in O.R., in: Journal of the Operational Research Society, Vol. 37, 1986, S.657 ff.

(67) Primio, F. di, G. Brewka, BABYLON: Kernsystem einer integrierten Umgebung für Entwicklung und Betrieb von Expertensystemen, in: Nachrichten für Dokumentation, 36. Jg., 1985, Nr.1, S.33 ff.

(68) Puppe, F., Expertensysteme, in: Informatik Spektrum, H. 9, 1986, S.1 ff.

(69) Rentrop, N., Ausgewählte Strategien im Gründungspro zeß, Reihe: Gründung, Innovation und Beratung, Hrsg. N. Szyperski, Bd. 2, Bergisch Gladbach, Köln 1985.

(70) Rentrop, N., Jetzt selbständig machen, München 1986.

(71) Rich, E., Artificial Intelligence, in: S. Shapiro (Hrsg.), Encyclopedia of AI, New York u.a. 1987, S.9 ff.

(72) Roesner, H., Expertensysteme für den kommerziellen Einsatz, in: S. Savory (Hrsg.), Künstliche Intelli genz und Expertensysteme, München, Wien 1985, S.35 ff.

(73) Roesner, H., S. Savory, Anforderungen an die Praxis, in: Expertensysteme, State of the Art 1/86, Hrsg. P. Schnupp, München 1986, S.26 ff.

(74) Savory, S., Artificial Intelligence State of the Art 1984, in: S. Savory (Hrsg.), Künstliche Intelli genz und Expertensysteme, München, Wien 1985, S.13 ff.

(75) Schieferle, D., Der praktische Einsatz in der Indu strie, in: Expertensysteme, State of the Art 1/86, Hrsg. P. Schnupp, München 1986, S.53 ff.

(76) Schierenbeck, H., Grundzüge der Betriebswirtschafts lehre, 5.Aufl., München, Wien 1980.

(77) Schmitz, P., A. Lenz, Abgrenzung von Expertensyste men und konventioneller ADV, in: Betriebswirtschaft liche Forschung und Praxis, H. 6, 1986, S.499 ff.

(78) Schreier, U., Die Beziehungen zwischen Datenbank und Expertensystemen, in: Expertensysteme, State of the Art 1/86, Hrsg. P. Schnupp, München 1986, S.30 ff.

(79) Shortliffe, E.H., Computer-based medical consulta tions: MYCIN, New York 1976.

(80) Sieben, G., W. Böing, R. Hafner, Expertensysteme zur Bewertung ganzer Unternehmen?, in: Betriebswirtschaft liche Forschung und Praxis, H. 6, 1986, S. 532 ff.

(81) Simon, H.A., The Structure of ill structured Problems, in: Artificial Intelligence, Vol. 4, 1973, S.181 ff.

(82) Smart, G., J. Langeland-Knudsen, The CRI Directory of Expert Systems, Hrsg. Learned Information, Ox ford, New Jersey 1986.

(83) Sowa, J., Semantic Networks, in: S. Shapiro (Hrsg.), Encyclopedia of AI, New York u.a. 1987, S.1011 ff.

(84) Stefik, M., Planning with constraints (MOLGEN: Part 1), in: Artificial Intelligence, Vol. 16, 1981, S.111 ff.

(85) Stück, H. H., So gründe ich ein Geschäft oder mache mich selbständig, 5.Aufl., München 1984.

(86) Swartout, W.R., Explanation, in: S. Shapiro (Hrsg.), Encyclopedia of AI, New York u.a. 1987, S.298 ff.

(87) Szyperski, N., Unternehmungsgründungen in der Kri sendynamik, in: Bratschitsch, R., W. Schnellinger (Hrsg.), Unternehmenskrisen - Ursachen, Frühwarnung, Bewältigung, Stuttgart 1981, S.149 ff.

(88) Szyperski, N., K. Nathusius, Probleme der Unternehmungsgründung, Stuttgart 1977.

(89) Trum, P., Automatische Generierung von Arbeitsplänen, in: Expertensysteme, State of the Art 1/86, Hrsg. P. Schnupp, München, Wien 1986.

(90) Volk, O.K., Expertensysteme für den betrieblichen Einsatz - Ansätze und Probleme, in: Betriebswirtschaftliche Forschung und Praxis, H. 6, 1986, S. 550 ff.

(91) Waterman, D.A., A Guide to Expert Systems, Reading (Mass.) u.a. 1986.

(92) Weizenbaum, J., Eliza, in: Communications of the Association of Computing Machinery (CACM), Vol.9, 1966, S.36 ff.

(93) Wilhelm, H., H. Corsten, P. Peckedraht, Erste Analyse ausgewählter Technologieparks in der Bundesrepublik Deutschland - Ergebnisse einer Befragung -, Zeitschrift für Betriebswirtschaftliche Forschung, 37. Jg. 1985, S.618 ff.

(94) Winston, P.H., B. Horn, Lisp, 2.Aufl., Reading (Mass). u.a. 1984.

(95) Witte, T., Heuristisches Planen, Beiträge zur industriellen Unternehmensforschung, Hrsg. D. Adam, Band 9, Wiesbaden 1979.

(96) Wolf, W., Wie mache ich mich selbständig?, München 1986.

(97) Yuhan, H., GPS, in: S. Shapiro (Hrsg.), Encyclopedia of AI, New York u.a. 1987, S. 323.

(98) Zeide, J.S., J. Liebowitz, Using Expert Systems: The Legal Perspective, in: IEEE Expert, Spring 1987, S.19 ff.

(99) Zelewski, S. von, Das Leistungspotential der Künstlichen Intelligenz, Bonn 1985.

(100) Zadeh, L.A., Fuzzy Sets, in: Information and Control, Vol. 8, 1965, S.338 ff.

(101) Zimmermann, H.-J., Optimale Entscheidungen bei unscharfen Problembeschreibungen, in: Zeitschrift für betriebswirtschaftliche Forschung, 27. Jg., 1975, S.785 ff.

Materialverzeichnis

(102) Bundesministerium für Wirtschaft (Hrsg.), Die ERP Programme 1986/87, Broschüre, Bonn 1986.

(103) Bundesministerium für Wirtschaft (Hrsg.), Förderungsmaßnahmen des Bundes für mittelständische Unternehmen, Freie Berufe und Existenzgründer, Broschüre, Bonn 1986.

(104) Bundesministerium für Wirtschaft (Hrsg.), Starthilfe, Ratgeber, Bonn 1985.

(105) Deutsche Ausgleichsbank (Hrsg.), Eigenkapitalhilfe, Broschüre, Bonn 1986.

(106) Dresdner Bank (Hrsg.), Informationen zur Existenzgründung, Broschüre und Arbeitsunterlagen, Frankfurt a.M. 1986.

(107) Dresdner Bank (Hrsg.), Selbständig mit Zukunftstechnologien, Broschüre, Frankfurt a.M. 1986.

(108) Industrie- und Handelskammer Nordrhein Westfalen (Hrsg.), Ich mache mich selbständig, Heft I + II, o.O. 1985.

(109) Norddeutsche Gesellschaft für Innovationsfinanzierung Innovationsfinanzierung, Werbeprospekt, Hannover 1986.

(110) Oldenburgische Landesbank (Hrsg.), Existenzgründung - Ihr überlegter Schritt in die Zukunft, Broschüre, o.O., o.J.

(111) Sparkasse (Hrsg.), Finanzierungshilfen für den Mittelstand, Broschüre, o.O. 1987.

(112) Stadt Münster (Hrsg.), Merkblatt: Existenzgründungsförderung, Münster o.J.

(113) Techno Venture Management, Eigenkapital für innovative Wachstumsunternehmen, Werbeprospekt, München o.J.

Tabellarischer Lebenslauf

Persönliche Daten	Geboren am 19.12.1960 in Bielefeld als Sohn des Kaufmanns Karl Freiherr von Wendt und der Hausfrau Hilke Freifrau von Wendt, geb. Heinemann	
Bildungsgang	1966 - 1969	Grundschule Olsberg
	1969 - 1978	Internat Stiftung Louisenlund
	12/78	Abitur
	4/79 - 7/80	Grundwehrdienst
	10/80 - 4/86	Studium der Betriebswirtschaftslehre an der Westfälischen Wilhelms-Universität Münster
	4/86	Examen als Diplom-Kaufmann
	5/86 - 2/88	Externe Promotion am Institut für Industrie- und Krankenhausbetriebslehre der Universität Münster (Leitung: Prof. Dr. D. Adam)
	3.2.1988	Mündliche Doktorprüfung durch die Wirtschaftswissenschaftliche Fakultät der Universität Münster